Journal of Advanced Transportation

Machine Learning, Deep Learning, and Optimization Techniques for Transportation 2021

Lead Guest Editor: Chi-Hua Chen
Guest Editors: Feng-Jang Hwang, Chunjia Han, Fangying Song, and Cheng Shi

Machine Learning, Deep Learning, and Optimization Techniques for Transportation 2021

Journal of Advanced Transportation

Machine Learning, Deep Learning, and Optimization Techniques for Transportation 2021

Lead Guest Editor: Chi-Hua Chen
Guest Editors: Feng-Jang Hwang, Chunjia Han, Fangying Song, and Cheng Shi

Contents

Contents

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 4390923, 16 pages
https://doi.org/10.1155/2022/4390923

Research Article

Automatic Scaling Mechanism of Intermodal EDI System under Green Cloud Computing

Qiang Huang,[1] **Lin Sun**,[1] **Furong Jia**,[1] **Jiaxin Yuan**,[1] **Yao Wu**,[1] **and Jinshan Pan**[2]

[1]*College of Information Engineering, Sichuan Agricultural University, Ya'an City, Sichuan Province 625014, China*
[2]*Southwest Jiaotong University, Chengdu City, Sichuan Province 610031, China*

Correspondence should be addressed to Jinshan Pan; jshpan@swjtu.cn

Received 5 April 2021; Accepted 3 April 2022; Published 30 May 2022

Academic Editor: Chi-Hua Chen

EDI is a hot topic in the research of multimodal transportation informatization, which determines the exchange level of intermodal transportation information. However, its high cost, large system coupling degree and low performance threshold cannot adapt to mass data exchange in high concurrent environment. Therefore, a decentralized, scalable, distributed and efficient data exchange system is formed. It plays a key role in realizing the comprehensive sharing of interdepartmental intermodal information in the cloud environment. In order to solve the problem of mismatching between application load and computing resource capacity and realize on-demand resource allocation and low carbon emission, this paper proposes to build an Extensible EDI system (XEDI) based on MSA and studies the scaling mechanism in container environment. Based on the resource scheduling characteristics of container cloud and considering the distribution and heterogeneity of intermodal cloud computing platform from the perspective of resource allocation, the automatic scaling mechanism of XEDI is established, the scaling model is established, and the automatic scaling algorithm is proposed. For Dominant Resource Fairness for XEDI (XDRF) resource allocation algorithm and Dominant Resource Fairness for XEDI (CXDRF) based on carbon considering energy consumption, the CXDRF algorithm is proved by quantitative experiments to achieve system performance optimization on the basis of ensuring system reliability and effectively reducing energy consumption. XEDI can not only meet the demand of dynamic load and improve service quality, but also reduce resource occupation and save energy by releasing virtual resources when resource utilization rate is low. It has great research significance and practical value for mass data application under low energy consumption conditions.

1. Introduction

Multimodal transportation is recognized as the most efficient mode of transportation service in the world, which is conducive to improving logistics efficiency and reducing logistics costs [1]. Among them, information sharing, as the key technology of the information of molten iron and intermodal transportation, not only determines the development level of the information of intermodal transportation, but also provides information guarantee for the realization of one-stop intermodal transportation service. China has established an intermodal transportation information sharing platform centered on some large ports, which has preliminarily realized data exchange between ports and railway departments, reduced cargo storage time and transportation cost, and effectively improved the efficiency of collaborative operation. Information sharing is the core of intermodal transportation informatization. Because the current information sharing technologies are all "chimney" architectures, they can only solve the problem of information integration in a local range and cannot meet the demand of on-demand information sharing in the cloud environment.

At present, the mainstream research is still the traditional EDI technology. Based on a research project conducted at the Institute of Logistics and Warehousing, Debicki and Kolinskianalyzed the impact of EDI methods on the complexity of information flow in global supply chains [2]. However, the traditional EDI technology has some problems, such as high cost, backward technology, and large coupling degree of the system, and the author does not provide corresponding solutions. Betz et al. applied ICT and

introduced the current application technology, connection type, message standard, and its impact on multimodal transport supply chain based on the international research results of Hamburg Port and Logistics Institute [3]. However, this technology is mainly customized for different users, and it is difficult to adapt to large-scale intermodal transportation systems. Ding explores the functions and operating conditions of relatively independent information systems for railways and ports, combined with traditional information exchange modes, and establishes an electronic platform suitable for information interconnection and intermodal station interoperability [4]. However, the traditional information exchange mode is still adopted, which cannot adapt to the massive data exchange in a high-concurrency environment.

At present, information sharing-related technologies adopted by core intermodal transportation organizations such as ports and railways mainly include the following:

1.1. Electronic Data Interchange. It refers to the formation of structured document data in accordance with relevant standards and the completion of end-to-end electronic data transmission methods. EDI standardizes and formats exchanged information in accordance with agreed protocols (such as EDIFACT and SOAP). It exchanges data between the computer network systems of trading partners through data transmission systems such as mail servers, FTP, and Message Queue (MQ), which can effectively solve the inefficiency of paper-based information transmission.

1.2. Service Oriented Architecture. It is defined as a functional paradigm for integrating dispersed businesses within an enterprise, and its essence is Enterprise Application Integration (EAI) technology that realizes information exchange between heterogeneous systems. The SOA component model realizes business information interaction between heterogeneous systems by defining standardized interfaces between different services and has the characteristics of loose coupling, coarse-grained, and transparency. As a technical realization of SOA, WebService has better openness and decoupling than traditional EDI.

1.3. Enterprise Service Bus. It is a bus-based enterprise-level SOA architecture with features such as interoperability, independence, modularity, and loose coupling. ESB takes services as the basic unit, and services are coordinated through messages to complete related business collaboration, and service consumers do not need to know the technical details of the service provider. ESB can not only reduce the workload of development and maintenance, save costs, and improve the scalability of the system, but also better deal with the heterogeneity of different technologies and protocols.

According to the research of relevant literature, the current application scope, advantages, and disadvantages of information integration technology in intermodal transportation informatization are shown in Table 1.

This paper constructs a self-scaling mechanism for the K8S-based XEDI (Extensible EDI) closed-loop control system, establishes an expansion model, and proposes an automatic expansion algorithm, a resource allocation algorithm, and a resource allocation optimization algorithm considering energy consumption to achieve flexible data sharing and on-demand resource allocation in a cloud environment. The performance of EDI system limits the energy consumption. Finally, the scalability test verifies that the proposed algorithm has good scalability effect and high scalability efficiency under heterogeneous cluster conditions, which can not only ensure the reliability of the system and realize the performance optimization of the system, but also effectively reduce energy consumption. It can not only meet the demand of dynamic load and improve the quality of service, but also reduce resource occupation and save energy by releasing virtual resources when resource utilization is low. It can solve the problems of high cost, system scalability, and insufficient data processing capacity of existing EDI system solutions.

This article is divided into eight parts:

(1) Introduction. This part generally introduces the methods used in this paper, also compares other methods, and discusses their advantages and limitations.

(2) The Introduction of XEDI. This section gives a detailed introduction to XEDI, including its advantages over EDI and the architecture of XEDI.

(3) Scaling models of XEDI. This part introduces the single-index scaling model of XEDI.

(4) Multi-index scaling model. This part introduces the multi-index scaling model of XEDI.

(5) Algorithms. In this part, the scaling algorithm based on the scaling model is introduced.

(6) Evaluation of the algorithm and example. This section tests XEDI's scalability performance.

(7) Conclusions. This part is a summary of the full text, and the performance of the algorithm in the article is summarized.

(8) Prospect. This section introduces the prospect of the algorithm and other application scenarios.

2. The Introduction of XEDI

Cloud computing has been developed as one of the creative platforms that give dependable, virtualized, and adaptable cloud resources over the Internet. Intermodal transportation refers to the "carriage of goods by two or more modes of transport." Traditional system framework of the intermodal transportation is rigid and lacks information sharing [5]. However, cloud computing helps provide a new direction to solve these bottlenecks and realize the informatization of the intermodal transport.

Electronic Data Interchange (EDI) refers to a standard for exchanging business documents, such as invoices and

Table 1: Comparison of different information integration technologies.

Information integration technology	Scope of application	Advantage	Disadvantage
EDI	Widely used in port informatization for data exchange between institutions	EDI message standards are perfect, which can better meet business needs	The cost is high, the technology is backward, the system coupling is large, the performance of the remote call method is low, the performance threshold is low, and it cannot adapt to the massive data exchange in the high concurrency environment
WebService	Used for business system integration of some electronic ports and ports	Mature technology, low coupling, low cost, and easy implementation of SOAP data standards	Its essence is a Web-RPC system, and the SOAP-based remote call method has a low performance threshold, and the supported message types are limited
ESB	There are relatively few applications in the interoperability industry, which is more suitable for the complex internal environment	Complete system, with standard adapters and extensible interfaces, low development, maintenance, and management costs, and strong compatibility with heterogeneity issues	The structure is cumbersome, the scalability is poor, and the software and hardware requirements are high. If different protocols are uniformly converted into SOAP messages through the adaptor and then XML parsing, there will be more unnecessary format conversions, especially the processing efficiency of large data packets

purchase orders, in a standard form between computers through the use of electronic networks like the Internet. It is widely used in the information sharing mechanism of intermodal transport. However, as time goes by, there appear more and more defects of EDI, such as high powerful consumption and low performance threshold, which make it hard to adapt to the mass data exchange under the cloud computing environment [6]. In order to realize the elasticity of information sharing, which expands when faced with high concurrent information processing and vice versa, we have to build a lighter and more flexible EDI system, named XEDI system in our paper.

When and how does the system stretch specifically? Though Kubernetes (K8s), a mature open-source system for automating deployment, scaling, and management of containerized applications, provides an ideal platform for hosting various workloads, automated scaling of the cluster itself is not currently offered, and thus, it is necessary to rebuild an automated scaling model based on that [7, 8].

Definition 1. XEDI system is the lighter and more flexible EDI system that we build, which provides open messages all intermodal participating organizations through the cloud.

Definition 2. Dominant Resource Fairness for XEDI (XDRF) is an algorithm designed to allocate the resources of Pods more fairly and perform better in calculating.

Contributions: this work has the following key contributions:

(1) It built the XEDI system and stretching model of that.
(2) It presented the algorithm to realize the scaling progressing.
(3) It provided a comparative analysis between our algorithm and others.
(4) It evaluated the energy consumption of the cloud system.

Compared with traditional EDI technology, XEDI has the following advantages:

(1) Low cost and high concurrency. Adopt micro-service unit encapsulation. The message processing module is encapsulated through microservices, which can be flexibly scheduled in the container cloud environment and simplify the construction of the scaling mechanism to achieve high concurrent message processing with variable loads with minimal computing resource consumption.
(2) Support remote call. Use asynchronous message mechanism. The asynchronous message protocol adapter (Takia) is used to realize message reception and forwarding, and the high-performance distributed queue system (Kafka) is used to replace the inefficient remote call and folder delivery polling mechanism of the traditional EDI system.
(3) Good scalability. Adopt an extensible message processing module. Message processing should be modularized, by encapsulating different message type processing procedures into micro-service units, and configured and extended according to message types and access protocols.

Different from the traditional EDI system, XEDI does not deploy to each port but manages it in a unified manner under a resource support system, renting functions such as

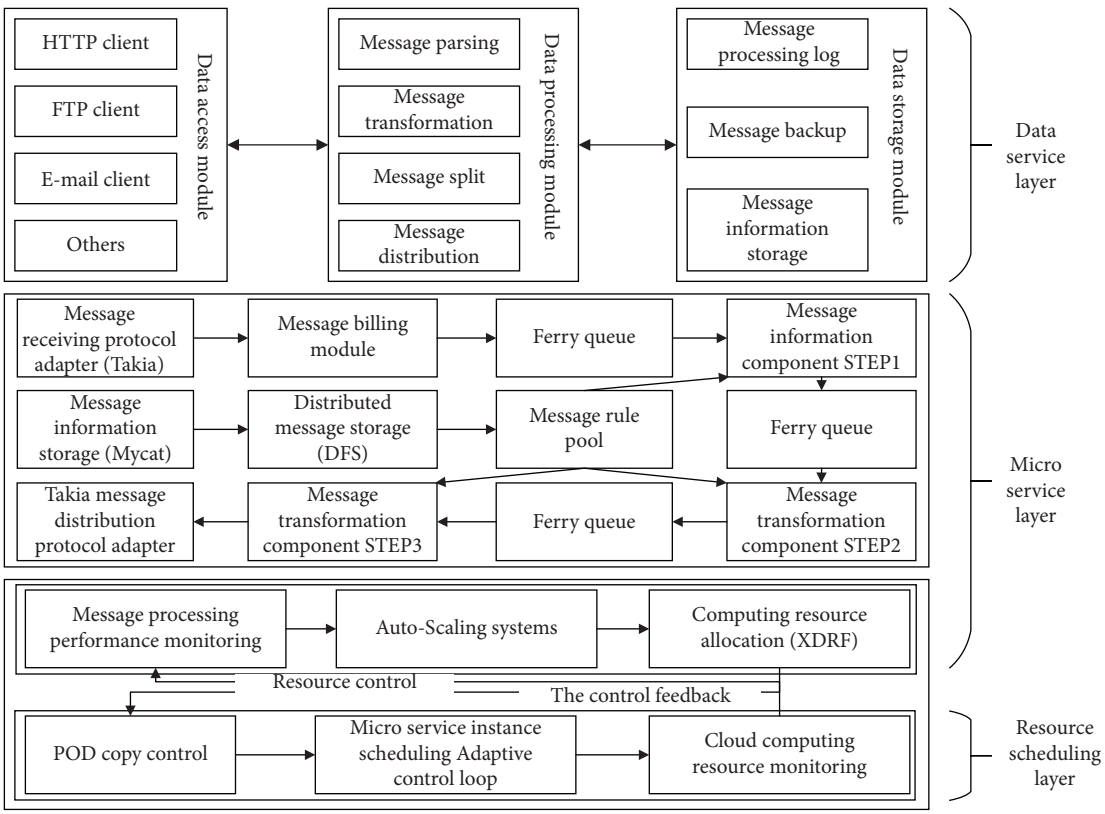


Figure 1: Hierarchical architecture of XEDI.

message exchange and distribution to participating institutions and users in the form of EDIaaS to reduce overall costs. However, the system design is mainly aimed at large-scale intermodal information platforms and has poor adaptability to the differentiated needs of individual users.

The construction of XEDI's architecture makes it clear about how the messages interact from different EDI systems under cloud. XEDI system are composed of Data Service Layer, Micro-Service Layer, and Resource Scheduling Layer from top to bottom. The logic structure of Data Service Layer is similar with the traditional EDI system. Considering business operation compatibility, Data Service Layer consists of three models: Data Access Modal, Data Processing Modal, and Data Storage Modal to make messages received and sent, parsed, and transformed and make messages stored [9]. To adapt containers scheduling, Data Processing Modal rebuilds the decentralized modal using micro-service. The last layer takes charge of component scheduling and the feedback of performance monitoring. The architecture is shown in Figure 1:

3. Scaling Models of XEDI

Most of the current scaling models and algorithms are designed based on IaaS VMS and can be divided into vertical and horizontal scaling modes [10]. However, it takes a long time to configure and start and stop virtual machine instances, so the scaling response is poor in real-time. Unlike IaaS, lightweight container clouds can scale applications in real time in a larger cluster environment. Because the container is an immutable carrier, only supporting horizontal scaling model, although the current container arrangement system has set up a simple response telescopic mechanism (for example K8S HPA [11]), but because only to copy an application based on memory and CPU load control, application scope is limited and has yet to have related research for complex component system scaling problem. Because XEDI's micro-service components are interconnected, there is no general scaling control by abstracting services into independent nodes [12]. In this paper, a closed-loop control system based on XEDI is proposed to build a self-scaling mechanism to achieve elastic data sharing and on-demand resource allocation in the cloud environment.

The scaling tactic is a function whose input is the indicator vector obtained from the XEDI monitoring module. Each dimension of the vector represents a monitoring indicator. In addition, the monitoring module ensures a long enough historical record. The record matrix P corresponding to the index data collection is shown as formula:

$$P = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1(m-1)} & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2(m-1)} & x_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{(n-1)1} & x_{(n-1)2} & \cdots & x_{(n-1)(m-1)} & x_{(n-1)m} \\ x_{n1} & x_{n2} & \cdots & x_{n(m-1)} & x_{nm} \end{bmatrix}. \quad (1)$$

N specifically represents the length of historical records, and m specifically represents the number of monitoring indicators.

The output of the scaling strategy is scaling index I, that is, $I = f_i(P)$, f_i is the scaling strategy function. If I is large, it can be interpreted as urgent to expand capacity and vice versa. In order to realize the quantization of the scaling decision, the system usually sets the expansion threshold I_{up} and the shrinkage threshold I_{down} in advance. If $I > I_{up}$ appears, the expansion process will be carried out at this time; if $I < I_{\text{down}}$, the shrinkage process will be carried out at this time.

According to the choice of P, f_i can be divided into single index strategy and multi-index strategy. When $m = 1$, f_i is a single-index scaling strategy; when $m > 1$, f_i is a multi-index scaling strategy. At the same time, according to the choice of f_i, the scaling strategy can be divided into response strategy and prediction strategy.

Although the single index algorithm is simple, it is prone to the miscalculation of scaling. In terms of scaling strategy, compared with a responsive scaling strategy, a predictive scaling strategy can make a prediction based on historical load and make scaling decisions earlier, which has a better scaling effect [13]. We propose a multi-index scaling model of XEDI based on a single index predictive scaling strategy.

In the single index algorithm, the input matrix P can be simplified as the historical window vector of load indicators, defined as follows:

3.1. Responsive Scaling Strategy. The nonprediction model is generally based on the historical window $H(x, n)$ to make a weighted average of the index x as the response value $V_r(H(x, n))$, where x is the index, and n is the window size. The following formula (2) is used for calculation:

$$V_r(H(x,n)) = \sum_{i=1}^{n} q_i x_i, x_i \in H(x,n), \tag{2}$$

where q_i is the weight coefficient. When $q_i = 1/n$, $V_r(H(x, n)) = V_{avg}(H(x, n))$ is the average value of indicator window $H(x, n)$. According to formula (3), the response scaling index I can be obtained:

$$I = f_i(P) = V_r(H(x,n)). \tag{3}$$

In particular, when $n = 1$, $I = x_n$, that is, scaling according to the current load, which is currently the industry commonly used scaling strategy.

3.2. Predictive Scaling Strategy. Compared with a responsive scaling strategy, a predictive scaling strategy can predict the historical load and make scaling decisions earlier, which has a better scaling effect [14]. In this paper, the autoregressive model is adopted to design a predictive scaling strategy, which is generally used in the stage of statistics and signal processing. As a random process, it is mostly used for modeling and forecasting various natural phenomena. Although XEDI message load changes are not the case. AR(p) specifically represents the p-order autoregressive model in this study. The definition of AR(p) model is as follows:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_t - i + \varepsilon_t. \tag{4}$$

The X_t is model variables, φ_i is the model of the regression coefficient, c is a constant (usually zero), ε_t is a random error, and p is the order number.

In the process of AR (1), a sliding window composed of multiple cycle monitoring indicators is used to predict the load value of future cycles, which are called adaptive Windows. According to $H(x, n)$ of the history window with length n, let the length of the adaptive window be w and iteratively predict the value of a new period based on n recent historical records. AR (1) can predict the index x_i of w future periods in the adaptive window, where $n < I < = n + w$, x_i is calculated iteratively by formula:

$$x_i = x_{avg} + \rho(1)\left(x_{i-1} - x_{avg}\right) + e_i, \tag{5}$$

where x_{avg} represents the mean value of x_i in the history window, e_i represents noise (generally 0), and $\rho(1)$ represents the relation function when the delay step number is *1*. $\rho(1)$ is calculated by the following formula:

$$\rho(l) = \frac{1}{n + w - l} * \sum_{i=1}^{n-l} \frac{\left(x_i - x_{avg}\right)\left(x_{i+l} - x_{avg}\right)}{\sigma_n^2}. \tag{6}$$

where σ_n^2 represents the standard deviation of the historical window.

Then, the predicted peak value can be obtained from the w window of indicator x_i. It is reasonable that when indicator x is the load rate, it can be calculated by the following formula:

$$V_p(H(x,n)) = \max\{x_i | i \in (n, n+w]\}, \tag{7}$$

The same as formula (3), the expansion index can be predicted $I = f_i(P) = V_p(H(x, n))$.

Although AR (1) algorithm solves the problem of index prediction in the time window, it can only realize the prediction of a single index load. Literature [15, 16] has proved through experiments that when the selection of indicators does not match with the type of load, the real load of the application cannot be shown, and even if the algorithm is rigorous, it will fail. The multi-index algorithm can rely on the comprehensive analysis of multiple load indicators to correctly judge the scaling time and effectively avoid the situation that the load is too large, the application scale cannot be adjusted correctly, and the request cannot be responded to normally.

4. Multi-Index Scaling Model

The basic idea of implementing the multi-index scaling strategy is to transform the multi-index load into a single index set. According to the above analysis, the input P of the scaling strategy is an $n * m$ matrix, and the output is the scaling index I. The multi-index scaling strategy is shown in the following formula:

$$I = f_i(P), x_{mn} \in P, m > 1, n \geq 1. \tag{8}$$

The calculation steps of I are as follows:

4.1. Convert Multiple Indicators into Single Indicators. The weighted average is carried out for each index in each row of P, and the transformation formula in the k row is as follows:

$$x_k = \sum_{i=1}^{m} x_{ki} * \frac{x_{ki}}{\sum_{j=1}^{m} x_{kj}}. \tag{9}$$

You take all the rows, you transform the input matrix P, and you get a vector that has dimension n, which is $H(x, n)$.

4.2. A Single Index Is Used to Calculate the Load. After converting the multi-index matrix into a single index vector, the single index scaling model can be used to calculate the scaling index I and carry out the scaling decision-making process.

At the same time, in terms of index selection, XEDI adopts the multi-index comprehensive trigger strategy that can reflect the performance most directly, so as to avoid the failure of prediction algorithm due to the failure of CPU, memory, and other indirect indexes to reflect the real status of message processing. The multi-index predictive scaling algorithm can select the trigger point of scaling more effectively and effectively prevent excessive scaling operation when combined with the cooling time of scaling.

Based on [17] Dominant Resource Fairness (DRF) algorithm and Dominant Resource Fairness for XEDI (XDRF), which is an extension of the two above-mentioned algorithms, it is designed to allocate the resources of PODs more fairly and perform better in calculating.

Assuming that there are n available computing nodes in the current cluster operating environment of XEDI, each computing node has m resources in total, Q_k represents the performance evaluation score of node k, η_k represents the ratio of the performance evaluation score of node k to the average score, and T_k represents the resource type characteristics of node k, and the encoding is consistent with Definition 3:

Definition 3. (XEDI performance context). Parameter $XEDI.C = \{$XEDI performance index set $\cup$ XEDI resource I resource status index set$\}$ is the performance context of the current XEDI system.

$\partial_{i,k}$ represents the adaptation factor of *POD(i)* on machine k, $D_{i,j}$ represents the demand of a copy of *POD(i)* for resources of type j, with $D_i = D_{i,1}, D_{i,2}, D_{i,3}, \ldots, D_{i,m}$, S_i represents the dominant share of *POD(i)*, R_j^k represents the total amount of resources of type j on node k, and $Ru_{i,j}^k$ represents the number of resources of type j that *POD(i)* has been allocated on node k, $Rc_{k,j}$ represents the number of resources of type j on node k that can be allocated, and W_i represents the weight of calculating $POD(i)$. The calculation process is as follows:

(1) W_i of each POD weight requiring capacity expansion in the POP set is

$$W_i = \frac{V(H(POD.MEMinsR, n)) + V(H(POD.CPUinsR, n))}{2}. \tag{10}$$

(2) The ratio of the performance evaluation score of node k to the average score is

$$\eta_k = \frac{nQ_k}{\sum_{i=1}^{n} Q_i}. \tag{11}$$

(3) The adaptive factor and dominant share S_i of *POD(i)* on node k were calculated, and k was

$$\begin{cases} S_{i,k} = \dfrac{\max_{j=1}^{m}\left\{Ru_{i,j}^k / R_j^k\right\} * \partial_{i,k}}{W_i}, \\ \partial_{i,k} = a * \eta_k + (1-a) * \dfrac{POD(i).RTP\&T_k}{POD(i).RTP}; s.t. 0 \leq a \leq 1. \end{cases} \tag{12}$$

(4) Calculate the leading share S_i (DS value) of *POD(i)* as the sum of its leading shares on each node:

$$S_i = \sum_{k=1}^{n} S_{i,k}. \tag{13}$$

(5) The resource Predicates set of computing *POD(i)* is

$$N_{pre}(i) = \left\{k | k \in \{1, 2, \ldots, n\}; \forall j \in \{1, 2, \ldots, m\}; D_{i,j} \leq Rc_{k,j}\right\}. \tag{14}$$

(6) The JTH resource allocation to *POD(i)* is determined by the following Priority:

When j is odd, the copy of *POD(i)* is allocated with suitable high-quality resources, as shown in the following formula:

$$s.t. k \in N_{pre}(i); \frac{\partial_{i,k} - a\eta_k}{1-a} > 0. \tag{15}$$

When j is even, the copy of *POD(i)* is allocated with suitable inferior resources, as shown in the following formula:

$$s.t. k \in N_{pre}(i); \frac{\partial_{i,k} - a\eta_k}{1-a} > 0. \tag{16}$$

5. Algorithms

The automatic scaling algorithm of XEDI is designed based on the scaling model in Section 2, which mainly solves the problem of when the message processing module scales in the container cloud environment. According to the threshold of the scaling index, the scaling process is divided into two algorithms: Algorithm 1 is for expansion, and Algorithm 2 is for shrinkage. The scaling algorithm firstly obtains the monitoring data, and under the condition that

the performance is not abnormal, calculate the load index set and calculate the XEDI message workload [18]. If the message's expansion index exceeds the expansion threshold, it traverses all the packet processing packets in sequence and calculates the data load of the corresponding POD. If the expansion index of the data packet exceeds the expansion threshold, the POD replica set is expanded to improve data throughput. On the contrary, if the expansion index is lower than the reduction threshold, the POD replica set is scaled down to release resources, and under the premise of ensuring the concurrent processing performance of the message, the resource occupation is minimized (Algorithm1).

6. Fairness Analysis of XDRF and CXDRF Algorithms

In the process of cloud resources sharing, the efficiency and fairness of the allocation of resources are the most important properties, widely considering the encourage sharing, cheat blocking, no jealous, and Pareto efficiency as an important index of judging allocation mechanism, and the following XDRF algorithm in POD expansion process is used to further discuss the equality of the allocation of resources:

Theorem 1. *XDRF is incentive sharing*

Proof. if there are k PODs to expand, for any *POD (i)*, *POD (j)*, satisfying $i \neq j, i \leq k, j \leq k$, $POD(i), POD(j) \in$ collection (POP), if satisfies $S_i < S_j < \ldots < S_k$, then the allocation of *POD(i)* results in the amount of resources D_i, and the total amount of resources decreases as $R = R - D_i$. According to formulas (3)–(6), the increase of the used resource Ru_j^k will cause the DS value S_i of *POD(i)* to increase. While $S_i > S_j$, *POD(i)* stops allocating and allocates resources to POD(j) to minimize the DS values alternation of different POD. When the load falls back, each POD will call Algorithm 3 to release the excess resources to ensure the resource-share of other POD in order to guarantee the resource-share of the next expansion, and the proof is completed. □

Theorem 2. *XDRF prevents strategic operations*

Proof. suppose that there are two resources r_1 and r_2, and the total resources are R_1 and R_2 respectively; there are two computing tasks i and j, and their resource demand vectors are $D_i = d_{i,r1}, d_{i,r2}$ and $D_j = d_{j,r1}, d_{j,r2}$. If the following relationship exists, $(d_{i,r1}/R_1) > (d_{i,r2}/R_2)$, $(d_{j,r1}/R_1) < (d_{j,r2}/R_2)$, then the dominant resource of computing task i is r_1, and the dominant resource of computing task j is r_2. If x_i and x_j are the number of subtasks of calculation tasks i and j respectively, the x_i and x_j are calculated by the following formula:

$$\begin{cases} d_{r,r1} * x_i + d_{j,r1} * x_j \leq R_1, \\ d_{r,r2} * x_i + d_{j,r2} * x_j \leq R_2, \\ \left(d_{i,r1}/R_1\right) * x_i = \left(d_{i,r2}/R_2\right) * x_j. \end{cases} \tag{17}$$

Assume that POD(i) increases its dominant resource demand from D_i to D_i' in order to obtain more shares, and the dominant resource of POD(i) is r, and then $d_{i,r} < d_{i,r}'$; if the dominant resource of POD(j) is p, according to formula (17), it can be known that when capacity expansion is completed,

$(d_{i,r}/R_r) * x_i = (d_{j,p}/R_p) * x_j$, and $(d_{i,r}'/R_r) * x_i = (d_{j,p}/R_p) * x_j$, because $(d_{j,p}/R_p) * x_j$ keeps the same , so $(d_{i,r}/R_r) * x_i = (d_{i,r}'/R_r) * x_i$ comes out, with the contradiction of $d_{i,r} < d_{i,r}'$, and the proof is completed.

It indicates that POD cannot increase its allocation share by falsely reporting resource demand and cannot be deceptive in meeting resource demand. □

Theorem 3. *XDRF is free of jealousy*

Proof. assume that POD(i) is jealous of POD(j)'s resource quota; that is to say, POD(j)'s resource quota is larger than POD(i), and these resources are also needed by POD(i). If these resources are $r \in r_1, r_2, \ldots, r_m$, the following two situations should be considered:

(1) If r is the dominant resource of POD(i) and POD(j), then r can only be the same resource. According to the hypothesis $d_{i,k} < d_{j,k}$ and according to formula (17) $(d_{i,r}/R_r) * x_i = (d_{j,r}/R_r) * x_j$, then $x_{i>}x_j$, that is, by allocating more copies for POD(i) to balance its dominant resource, so the resource allocation of i will not be affected.

(2) If r is not the dominant resource of i but is relatively important for POD(i), POD(j) occupies more quotas, and if the dominant resource distribution of POD(i) and POD(j) is q and p, then there is the following relationship:

$(d_{j,p}/R_p) * x_j = (d_{i,q}/R_q) * x_i > (d_{j,r}/R_r) * x_j > (d_{i,r}/R_r) * x_i$, consider the following two scenarios simultaneously:

(a) if $(d_{j,p}/R_p) > (d_{i,q}/R_q)$, then $x_i > x_j$, and in order to satisfy the above relationship, the demand of POD(i) on r is far less than that of POD(j), that is, $d_{j,r}?d_{i,r}$, and r is not an important resource of POD(i), which contradicts the hypothesis.

(b) if $(d_{j,p}/R_p) < (d_{i,q}/R_q)$, then $x_i < x_j$, and it can be obtained from the above relationship, $d_{j,r} \geq d_{i,r}$, which is the same as the case a), so the demand of POD(i) on r is less than or equal to that of POD(j), which is inconsistent with the hypothesis, and the proof is completed. □

Theorem 4. *XDRF is satisfy Pareto efficiency*

Proof. according to the definition of Pareto efficiency, it is assumed that POD can increase its quota without affecting the quota of other pods. According to the hypothesis, for POD(i), Pareto improvement exists to make it increase the resource share of POD(i) without affecting the share of other nodes. According to lemma (8) in literature [19], there is at least one saturated resource in POD using DRF. Suppose that the share

```
name: autoScalingUp
input: none
output: none
Define variable I: Expansion index; Define variable scalingStrategy: Capacity expansion policy: Non-predictive capacity expansion if
the value is 0, predictive capacity expansion if the value is not 0; Define a collection<POD>: POD collections that need to be
expanded; Define a collection<EXPOD>: A collection of pods with poor performance; Define a collection<POP>: POD optimization
solution set;
  Main-loop {
    retrieve XEDI context from CAT as XEDI.C; //Get the XEDI context from CAT
    //Calculate the average message processing time and average throughput of XEDI
    XEDI .T_avg(n) = V_avg(H(XDE I.T_ins,n)); XEDI .V_avg(n) = V_avg(H(XDE I.V_ins,n))
    if (XEDI .T_avg(n) > XEDI .T_max and XEDI .V_avg(n) < XEDI .V_max) {
      //POD nodes with normal throughput but abnormal message processing time
      for (step i from 1 to 3) {
        for (each XEDI pod in step i from K8S) {
          retrieve pod. C;
          // Calculate the average data processing time and average throughput of POD
          POD.T_avg(n) = V_avg(H(POD.T_ins,n)); POD.V_avg(n) = V_avg(H(POD.V_ins,n))
          if (POD.T_avg(n) > POD.T_max and POD.V_avg(n) < POD.V_max) {
            add this unhealthy pod to collection<EXPOD>;
          }}}
        report collection<EXPOD> to CAT as performance exception; //Report exception triggers to CAT
        enter next loop;
      }
      If (scalingStrategy = = 0) {
        //According to formulas (2) and (3), q_i = 1, the responsive capacity expansion index based
        //on mixed load rate is calculated
        I = V_avg(H(XEDI .TVinsR,n));
      } else {
        //According to formulas (5)–(7), the predictive expansion index was calculated
          I = V_p(H(XEDI .TVinsR,n));
      }
      if (I > I_max) {
        //Enter the expansion process and get the POD set of XEDI
        for (each XEDI pod in K8S) {
          retrieve pod. C;
          //Avoid frequent POD scaling by cooling-off time
          if (currentTime-pod. lastScalingTime < pod. C. T_c) {
            enter next loop;
          }
          //Calculate POD expansion index
          If (scalingStrategy = = 0) {
            //According to formulas (2) and (3), q_i = 1, the response expansion index based on
            //mixed load rate was calculated
            I_p = V_avg(H(POD.TVinsR,n));
          } else {
            //According to formulas (5)–(7), the predictive telescopic index is calculated
            I_p = V_p(H(POD.TVinsR,n));
          }
          //Make scaling decisions
          if (I_p > I_max) {//Calculate and update POD expansion metrics
            If (scalingStrategy = = 0) {
              //In response mode, q_i = 1 is calculated according to formulas (2) and (3) to calculate //data grouping processing
time and throughput,
              //Queue wait time and response metrics for CPU and memory utilization
              update pod context (
V_avg(H(POD.Tins,n)),V_avg(H(POD.Vins,n)),V_avg(H(POD.Qins,n)),V_avg(H(POD.MEMinsR,n)),V_avg(H(POD.CPUinsR,n)));
            } else {
              //Predictive mode, according to formulas (4)–(6), calculate data grouping //processing time, throughput, Queue
wait times and predictors of CPU and //memory utilization
              update pod context (
```

Algorithm 1: Continued.

```
V_p(H(POD.Tins,n)),V_p(H(POD.Vins,n)),V_p(H(POD.Qins,n)),V_p(H(POD.MEMinsR,n)),V_p(H(POD.CPUinsR,n)),);
            }
            add this pod to collection<POD>;
        }}
        //Calculate configuration optimizations for POD collections that need to be scaled up
        for (each pod in collection<POD>) {
            //POD optimization scheme is calculated by queuing theory system
            compute PodOptimizationPlan(pop) for pod by queue system;
            add this pop to collection<POP>;
        }
        //Confirm whether K8S resources meet the expansion conditions
        if (R not adequate for collection<POP> scaling-up) {
        //When the available resources are used up, try to apply for resources from the container
        //cloud and preempt dynamically when the resources are insufficient try to apply resource
        //increment as R_c1;
            if (R_c1 == 0) {
                //Performance is abnormal and additional computing resources cannot be
                //requested from the container cloud report resource exhausted exception to
                //CAdvisor;
                    enter next loop;
            }
        R_c = R_c1; R = R + R_c; //Update total resources
            }
            //Allocate resources for POD according to Algorithm2
            call XDRF algorithm for collection<POP> with Rc By XTuning. Scheduler;
        }
        If (I < I_min) {
            //Normal performance without expansion, according to the Algorithm 3 asynchronous
            //trigger shrinkage process
            asyn_invoke auto scaling-down with XEDI.C;
        }}
```

Algorithm 1: XEDI's automatic expansion algorithm.

of POD(I) in resource r is increased from $s_{i,r}$ to $s'_{i,r}$, where $s_{i,r} = d_{i,r} * x_i$. According to Theorem 2, POD(I) cannot increase $s_{i,r}$ by increasing $d_{i,r}$; therefore, POD(I) can only increase the quota of resource r by increasing x_i. It can be obtained from lemma (8) that I has at least one saturated resource w. Therefore, increasing xi cannot increase the share of w. Therefore, it contradicts the hypothesis that Pareto improvement does not exist, and the proof is completed.

The essence of XDRF meeting Pareto efficiency is the constraint on resource occupation by P. When resource allocation reaches saturation, POD cannot increase its share anymore, unless it occupies resources of other pods, whose behavior will be forbidden by XDRF.

The XEDI system is deployed with the help of InterThings, a virtual cluster environment of containers, and it is tested under the following two aspects: scaling effect and scaling velocity [20]. The former one refers to comparing frequencies of message processing using different automatic scaling algorithms, while the latter one refers to testing whether PODS can be effectively adjusted with the change of load [1]. □

7. Scalability Test of the Algorithm

7.1. Scaling Effect Test. The throughput limit and resource allocation algorithm efficiency of XEDI under different POD copy sizes were tested, among which the Takia adapter was configured into SYN mode; that is, the request-response was not conducted until the message conversion of the three steps was completed. The POD copy quota of the three steps tested was configured as <0.2c, 128M>, <0.4c, 128M>, <0.3c, 256M>, and the resource vectors were <1, 0, 0>, <0, 1, 1>, and <1, 0, 1>. In order to compare the capacity expansion effect, XEDI configured the three STEP copies of the test Topic into even capacity expansion mode (from 2 to 16 to expand capacity on 4 heterogeneous computing nodes), where Dell-R710 and Dell-R620, respectively, correspond to CPU and memory storage computing resources and used Mesos DRF and XDRF as XEDI POD allocation algorithms, respectively [21].

The VUser of LoadRunner adopts the trapezoid incremental graph until the HTTP-503 error appears in the response result. Thus, the response frequency of server requests, data throughput frequency, and maximum concurrent request number of XEDI under different replica configurations can be obtained, as shown in Table 2:

And the relationship of the data in Table 2 can be shown in Figures 1–4.

According to Figure 4, through XMON's monitoring of POD's comprehensive load rate, the overall load rate of XDRF is higher than that of Mesos' DRF algorithm during the POD distribution process, which indicates that the resources are

```
name: autoScalingDown
input: C: XEDI performance context;
output: none
  //If the XEDI resource occupancy rate is low, it will not shrink, reducing the number of
  //unnecessary shrinkages
  if (R_a < R * θ_max) { terminate scaling-down;}
  //Get all the POD sets for XEDI
  retrieve all pods of XEDI from K8S as collection<POD>;
  for (POD pod: collection<POD>) {
    retrieve pod. C;
    //Avoid frequent POD scaling by cooling-off time
    if (currentTime-pod. lastScalingTime < pod. C. T_c) {
      enter next loop;
    }
    /*Calculate POD expansion index*/
    If (scalingStrategy == 0) {
      //According to formulas (2) and (3), q_i = 1, the response capacity expansion index based
      //on mixed load rate was calculated
      I = V_avg(H(PO D.TVinsR,n));
    } else {
      //According to formulas (5)–(7), the predictive expansion index was calculated
      I = V_p(H(PO D.TVinsR,n));
    }
    //Adjust pods with lower load rates
    if ((I < I_min) {
      //Reduce the number of copies of POD according to the flex index
      pod. replicas = *I;
      //Refresh the POD's copy number configuration so that the POD's shrinkage takes effect
      refresh pod replication for K8S with XTuning. Scheduler;
    }
    //Stop shrinking when the resource utilization rate falls below the resource load rate
    if (R_a < R * θ_min) { terminate scaling-down;}
}
```

Algorithm 2: XEDI's automatic shrinkage algorithm.

better utilized overall. Combined with Figure 5 it can be also seen that XDRF algorithm and dynamic weighting and resource types match, and the more the urgent priority allocation, the more the reasonable resources, as well as the equilibrium between different node performances, so the two resources allocation performance is better than the default resource allocation algorithm as a whole.

7.2. Scaling Velocity Test

7.2.1. Comparison of Scaling Effects of Different Scaling Strategies. In the cloud deployment response scale and scale forecasting strategy, respectively, two cluster instances, Takia ferry mode is configured to ASYN enough throughput to ensure that the front end POD configuration is the same as the first step in the test, the initial replications to 1, and in the test phase of the load scenario, LoadRunner VUser adopts arch random graph, the two cluster instances at the same time to request access to 16 min to test the system's response to the load, including the expansion of the trigger and execution and time efficiency to solve this problem. According to the interface of the capacity enlargement algorithm, the capacity expansion threshold was set as 75% and the capacity reduction threshold as 45% [22]. The capacity expansion index adopted the time-throughput composite load rate, and the cooling time of capacity expansion was 2 min (note: in production environment, to avoid frequent capacity expansion caused by load fluctuation near the threshold, the value was generally more than 10 minutes). The test results are shown in Figure 6.

As can be seen from Figures 1 and 6, in the initial stage, the load rate is lower than 40%, and the total number of POD copies is specifically 3. At 3 min, the load increases sharply, and the server load rate rises rapidly to nearly 80%, higher than the capacity expansion threshold. With capacity expansion triggered, the number of POD copies increased to 8 at 4 min. After that, the load was reduced to 40%, less than the shrinkage threshold. Since it was in the "cooling-off time" stage, the shrinkage operation was not triggered, and the two expansion and shrinkage operations within a short period of time in this stage were prevented. At 5 min, the load returned to the rising trend, reached 75% at 7 min, and triggered the second capacity expansion operation. At 8 min and 9 min, respectively, the number of copies of the two expansion strategies increased to 12. At 11 min, the load was reduced, and the volume reduction operation was triggered when it was lower than the volume reduction threshold. The copies of the two capacity

```
Algorithm name: XDRFforPOD
/*The number of nodes is n, and the resource dimension is m*/
Input: R = <R_1 = <R_{1,1} to R_{1,m}>, . . ., R_k = <R_{k,1} to R_{k,m}>, . . ., R_n = <R_{n,1} to R_{n,m}>>: total resource collection; collection<POP>: POD
optimization scheme collection;
Output: none
Define variable z = collection<POP>.size: the number of PODs to be calculated; Define variable R_{1u} = Ru^1_{1,1}, . . . , Ru^k_{i,j}, . . . , Ru^n_{z,m}: the
set of allocated resources, Ru^k_{i,j} represents the number of resources of type j that has been allocated by POD(i) on node k; Define
variable R_c = Rc_{1,1}, . . . , Rc_{k,j}, . . . , Rc_{n,m}: unallocated resource set, Rc_{k,j} represents the number of resources of type j on node k that
can be allocated; Define variable W = W_1 to W_z: The weight set of POD to be optimized;
  for (i from 1 to z) {
    Calculate the weight of the POD in the collection<POP> according to formula (10) and fill the collection W;
}
  for (k from 1 to n) {
    Calculate the cluster nodes η_k according to formula (10) and arrange them in ascending order;
  }
  do {
    for (i from 1 to z) {
    For R and Ru sets, calculate the dominant share S_i of POD(i) according to formulas (11) and (12), and update the
collection<POP> collection, sorted by S_i in ascending order;
  }
  //Get the POD with the smallest dominant share (i)
  picking POP(i), the first element of collection<POP>;
  POD(i) = POP(i).POD;
  //Get POD(i) resource requirements such as CPU and memory
  calculate resource demand of POD(i) as D_i;
  Calculate the resource Predicates set Npre(i) of POD(i) according to formula (14);
  if (Ru + D_j ≤ R) {
    According to formulas (13) or (14), a copy resource r is allocated to POD(i), where r = = D_i;
    //Load and run the copy instance
    let replication as result of loading and running POD(i).replicationConfig with r;
    //Register the copy, monitor the data queue and participate in data processing services
    register this replication as consumer to kafka with POP(i).topic;
    //Update resource usage
    Ru+ = D_j; Rc− = D_i;
    //Refresh the dominant share of POD(i) according to formulas (12) and (13)
    refresh dominant share for POD(i);
    if (POP(i).dPR-- == 0) {
      //The POD has been expanded and deleted from collection<POP>, and no longer enters //the subsequent allocation process
      POD(i) scaling-up done;
    }}
  //The cluster node resources are exhausted, record the POD information that has not been
  //allocated and exit DRF
  else {
    get unsatisfied POPs as collection<UPOP> from collection<POP>;
    report collection<UPOP> to CAdvisor;
    terminate XDRF;
  }
  / When collection<POP> is empty, all POD allocation is completed
  If (allocation done for all pod in collection<POP>) {
    report to XTuning allocation done with R and collection<POP>;
    terminate XDRF;
  }} while (true)
```

Algorithm 3: XDRF algorithm.

expansion strategies were reduced to 6 and 8, respectively, and the volume expansion was not carried out in the following 2 min cooldown. At 16 min, the volume reduction operation was triggered by load drop, and the number of copies was reduced to 3, which verified the effectiveness of XEDI dynamic capacity expansion. It can also be seen from the figure above that, compared with the response capacity expansion strategy, the predictive capacity expansion strategy is more active than the response strategy, because it can predict the load status of the subsequent time series in advance. Therefore, the capacity expansion preparation can be carried out before the load expansion, so as to obtain better system processing performance and throughput [23].

Table 2: Comparison of pressure test results before and after replica expansion.

Number of copies	Index					
	Request response frequency (fetches/sec)		Data throughput frequency (bytes/sec)		Maximum number of concurrent requests	
	DRF	XDRF	DRF	XDRF	DRF	XDRF
2	0.1624	0.1645	32.5	33.4	60	64
4	0.3056	0.3742	65.4	66.8	113	137
6	0.4475	0.5238	85.2	87.1	152	174
8	0.5834	0.6925	112.4	114.5	184	206
10	0.7423	0.8292	121.7	124.3	216	268
12	0.7893	0.9482	137.6	139.7	262	338
14	0.8621	1.0179	142.8	145.2	287	377
16	1.0016	1.1382	146.9	150.4	321	406

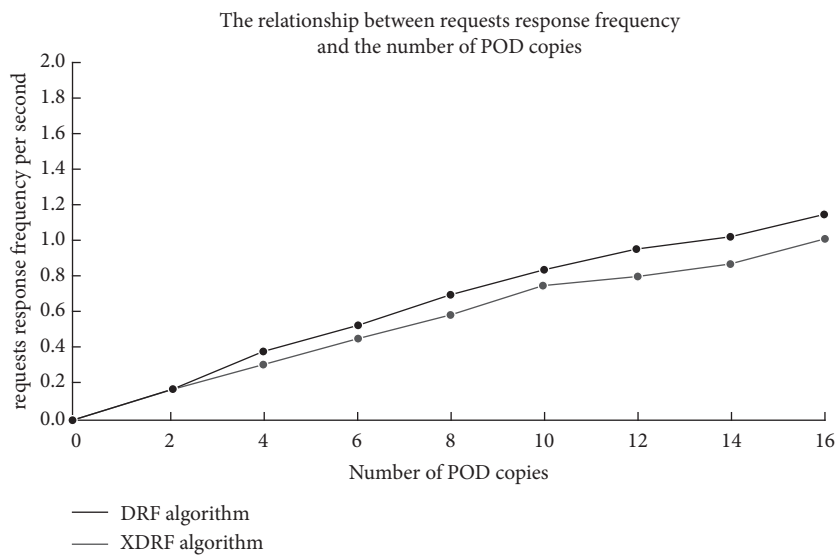


Figure 2: The relationship between XEDI request-response frequency and the number of POD copies.

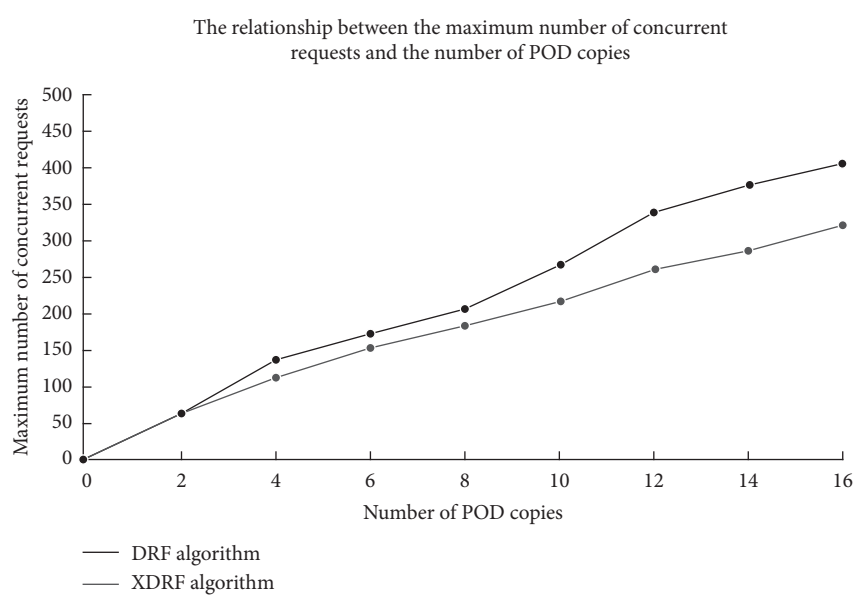


Figure 3: The relationship between the maximum number of XEDI concurrent requests and the number of POD copies.

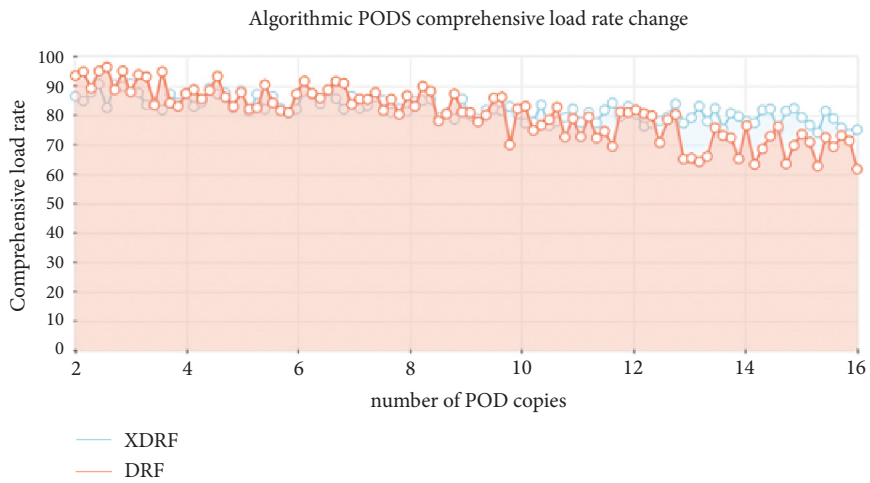


Figure 4: XEDI-PODS CPU-RAM comprehensive load rate change trend.

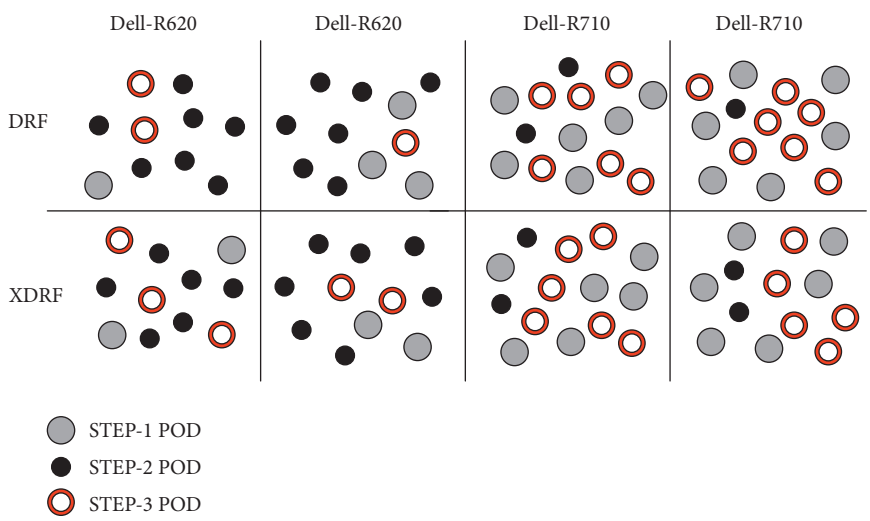


Figure 5: Resource allocation results of different resource allocation algorithms under the same load.

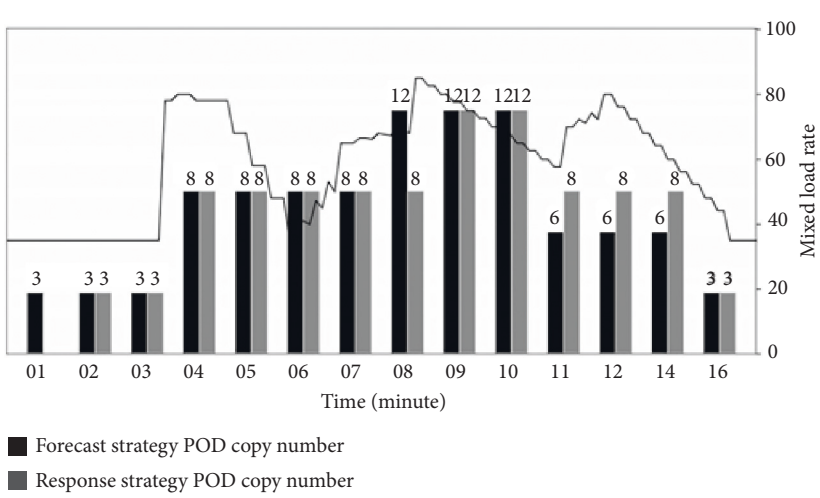


Figure 6: Comparison diagram of automatic capacity expansion effect of XEDI under different strategies.

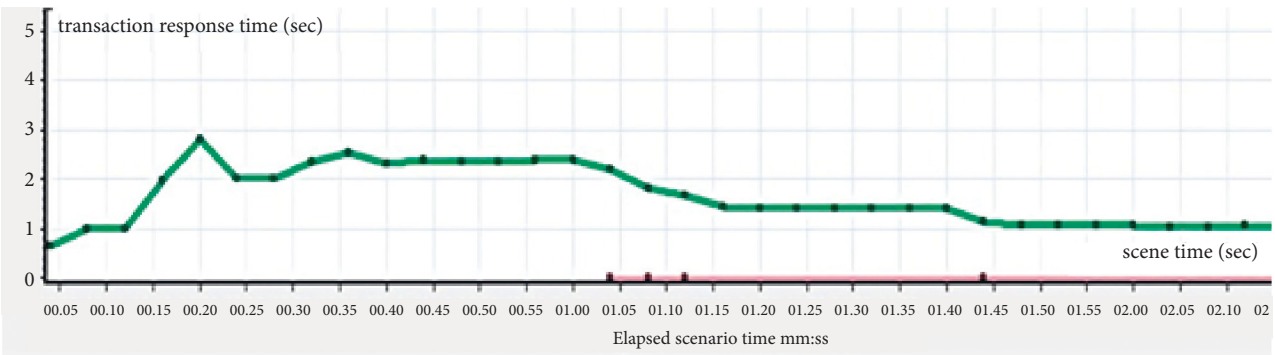


Figure 7: Response spectrum of automatic expansion system.

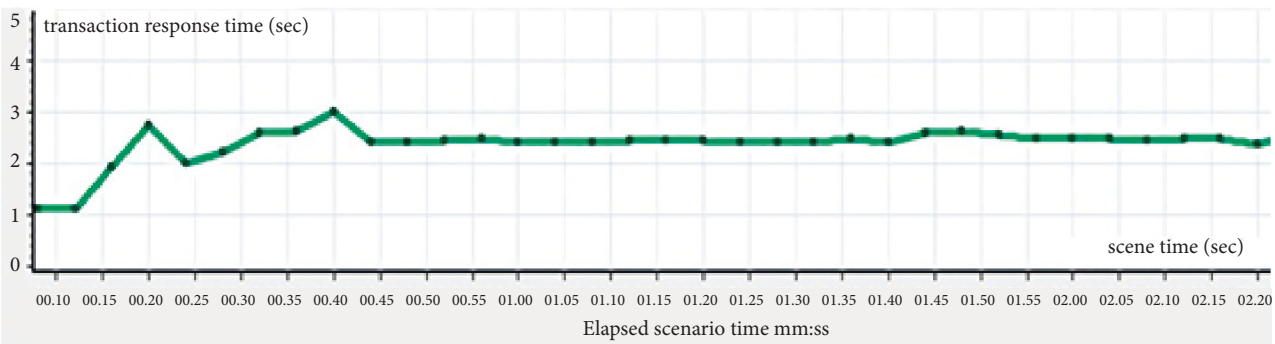


Figure 8: Response spectrum of nonexpandable system.

7.2.2. Performance Comparison between Closed and Open Scaling Strategy. Based on the above test scenarios, and further comparison does not have scale characteristics of the traditional "stovepipe" through information sharing system with elastic performance difference between unit ITIU information sharing, namely, validation expansion module performance improvement effect of information sharing service, we will have the response type expansion cluster instance XTuning closed, as well as the expansion and test instance and the expansion of the client's response performance. Using the same server configuration, set the front module to the SYN mode, and at the same time, set the LoadRunner VUser map of 300 concurrent users trapezoidal map; the threshold arrival time is 50 sec, cycle for 3 min, and do not test points recording two-cluster-instance transaction response time, and the test results are shown in Figures 7 and 8; the *X*-axis is time, the vertical axis for the transaction response time.

By comparing the two figures, it can be found that the response time of the two cluster instances is basically the same in the early stage, and the system throughput of the server load reaches the threshold at about 50 sec. In Figure 7, as the capacity expansion scenario starts to expand, the message response time decreases to about 1 sec after the capacity expansion. In Figure 8, as the cluster instance shuts down the capacity expansion component, the response time of the system after stabilization remains around 2.5 sec. It can be seen that the automatic capacity expansion system can effectively maintain the service performance of the client when the system load increases.

In order to compare the performance difference of XEDI components in the container environment and the virtual machine environment of the current mainstream cloud platform, two XEDI cluster instances that respond to the scaling mode are deployed in the container and virtual machine environments [24, 25]. The Takia adapter is configured in SYN mode, the POD configuration is the same as the first test, the initial number of copies is 1, the node in the virtual machine mode also uses the same configuration, and the initial number of nodes is also 1. LoadRunner's VUser map is a ladder map of 200 concurrent users, with a period of 2 minutes. Record the transaction response time of the two test cluster instances separately to evaluate the response level of the container and virtual machine scaling to the load under the same configuration. The test results are shown in Figures 9 and 10.

As can be seen from the above figure, the average response time of the container environment is about 1 sec, which is significantly better than the virtual machine environment. In addition, because the container is a lightweight process-level service, the refresh time of the POD copy only takes about 5 sec, so Figure 9 can quickly complete the expansion operation in the early stage of the load and reduce the transaction response time to less than 1 sec. The virtual machine startup and deployment time is an operating system level operation. As can be seen from Figure 10, the transaction response time in the virtual machine mode increases as the load reaches about 90 sec before completing the first expansion operation. It can be seen that, in terms of

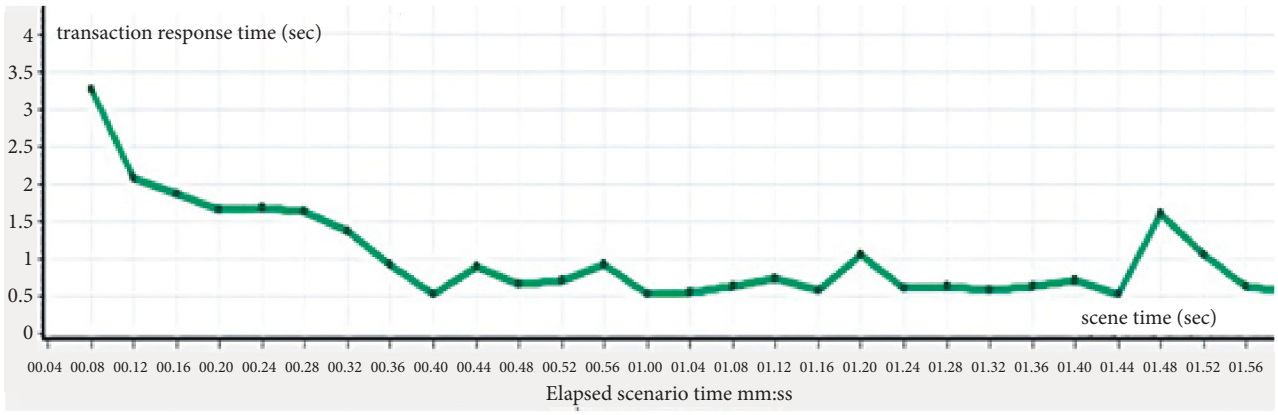


Figure 9: Container mode expansion deployment response time.

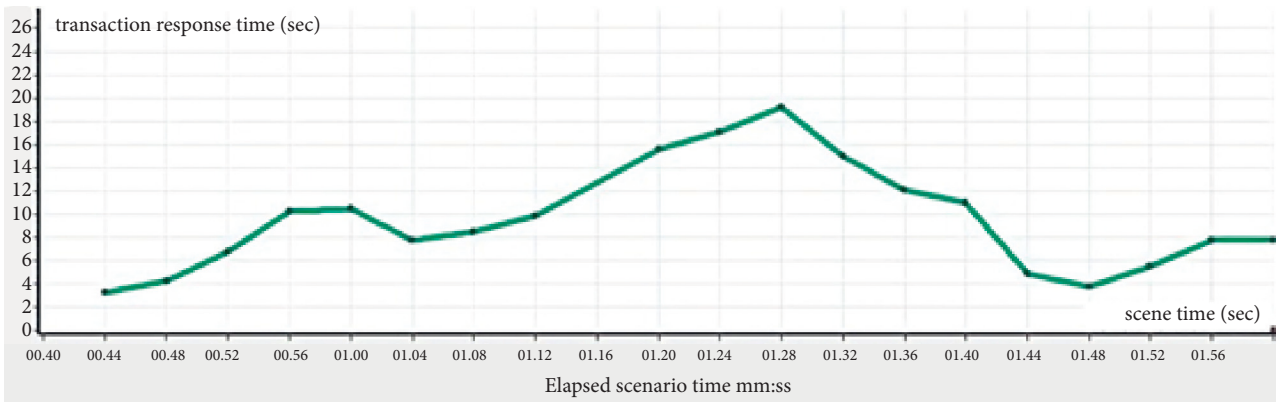


Figure 10: Response time of expansion deployment in virtual machine mode.

scalability and agility, container clusters have obvious advantages over virtual machines.

8. Conclusions

In this paper, we have proposed the autoscheduling algorithm XDRF in the cloud environment. This paper incorporates a detailed evaluation of the XEDI stretching model toward the workloads of CPU and RAM. Through quantitative experiments, it was verified that the XDRF algorithm could achieve the system performance optimization on the basis of guaranteeing system reliability and reduce energy consumption effectively [26]. The work in this paper also has clarified that the model can meet the demand of dynamic load and improve the service quality according to the two tests.

9. Prospect

9.1. Standardization of Cloud Platform for Combined Iron and Water Transport. Cloud computing is an effective way to optimize the existing intermodal information layout and application management model, and it also brings new challenges to intermodal business and data standards under the cloud environment. Although the intermodal cloud platform adopts a centralized management model, it is difficult to integrate a large number of heterogeneous intermodal applications on a unified cloud platform without a unified intermodal information standard. Although simple migration can achieve unified management of applications, it cannot effectively use virtual resources to optimize cloud service models. Therefore, researching the intermodal information standards that adapt to the cloud environment is crucial to the landing application of intermodal cloud platforms.

9.2. Construction of Intermodal Blockchain. Combined transportation of iron and water is a multiparty collaborative business process, and the security and traceability of information sharing are extremely important. Blockchain is the latest information sharing and storage technology. It can not only effectively simplify the intermodal business process, but also effectively protect the security of shared data. How to combine blockchain with intermodal information technology, build intermodal blockchain, and realize intermodal smart contracts and data traceability is also of great significance and requires a lot of follow-up research work.

Data Availability

Data used to support the finding of this study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Sichuan Agricultural University education reform project (X2013039 and X2014025) "Agricultural Information Engineering" Sichuan key laboratory of higher education; the National Key Research and Development Program fund (2017 yfb1200702 and 2016 yfc0802208); the National Nature Foundation Project (61703351); the Science and Technology Research Project of China Railway Corporation (P2018T001); and the Science and Technology Plan Project of Sichuan Province (2018RZ0078 and 2019JDR0211).

References

[1] Q. Lei, W. Liao, Y. Jiang, M. Yang, and H. Li, "Performance and scalability testing strategy based on kubemark," in *Proceedings of the IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp. 511–516, IEEE, Chengdu, China, 2019, April.

[2] C. C. Lo, K. M. Chao, H. Y. Kung, C. H. Chen, and M. Chang, "Information management and applications of intelligent transportation system," *Information Management and Applications of Intelligent Transportation System*, vol. 2015, Article ID 613940, 2 pages, 2015.

[3] T. Debicki and A. Kolinski, "Influence of EDI approach for complexity of information flow in global supply chains," *Business Logistics in Modern Management*, vol. 18, 2018.

[4] J. Betz, E. Jaskolska, M. Foltynski, and T. Debicki, "The impact of communication platforms and information exchange technologies on the integration of the intermodal supply chain," in *Integration of Information Flow for Greening Supply Chain Management*, pp. 131–141, Springer, Berlin, Germany, 2020.

[5] L. Ding, "Multimodal transport information sharing platform with mixed time window constraints based on big data," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1–11, 2020.

[6] S. Liu, C. Yin, D. Chen, H. Lv, and Q. Zhang, "Cascading failure in multiple critical infrastructure interdependent networks of syncretic railway system," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[7] T. Yang, X. Peng, D. Chen, F. Yang, and M. Muneeb Abid, "Research on trans-region integrated traffic emergency dispatching technology based on multi-agent," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 5763–5774, 2020.

[8] S. Taherizadeh and V. Stankovski, "Dynamic multi-level autoscaling rules for containerized applications," *The Computer Journal*, vol. 62, no. 2, pp. 174–197, 2019.

[9] L. Li, "Energy consumption management of virtual cloud computing platform," in *Proceedings of the IOP Conference Series: Earth and Environmental Science*, vol. 94, no. 1, Article ID 012193, 2017.

[10] D. Chen, S. Ni, C. A. Xu, and X. Jiang, "Optimizing the draft passenger train timetable based on node importance in a railway network," *Transportation Letters*, vol. 11, no. 1, pp. 20–32, 2019.

[11] A. Sun, T. Ji, Q. Yue, and F. Xiong, "IaaS public cloud computing platform scheduling model and optimization analysis," *International Journal of Communications, Network and System Sciences*, vol. 4, no. 12, pp. 803–811, 2011.

[12] T.-T. Nguyen, Y.-J. Yeom, T. Kim, D.-H. Park, and S. Kim, "Horizontal pod autoscaling in Kubernetes for elastic container orchestration," *Sensors*, vol. 20, no. 16, p. 4621, 2020.

[13] Z. Cao, *Research on Dynamic Scaling of Web Application Deployment in Cloud Platform*, Doctoral dissertation, Fudan University, Shanghai, China, 2012.

[14] A. Chandra, W. Gong, and P. Shenoy, "Dynamic resource allocation for shared data centers using online measurements," in *International Workshop on Quality of Service*, Springer, Berlin, Germany, 2003.

[15] J. Bi, Z. Zhu, R. Tian, and Q. Wang, "Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center," in *Proceedings of the IEEE 3rd International Conference on Cloud Computing*, pp. 370–377, IEEE, Miami, FL, USA, 2010, July.

[16] W. Ou and Q. Hu, "Modeling and simulation of CIMS logistics dispatching system," *Computer Integrated Manufacturing Systems*, vol. 10, no. 9, pp. 1067–1072, 2004.

[17] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: fair allocation of multiple resource types," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, vol. 11, p. 24, Massachusetts, MA, USA, 2011, March.

[18] H. W. Wang, C. H. Chen, D. Y. Cheng, C. H. Lin, and C. C. Lo, "A real-time pothole detection approach for intelligent transportation system," *Mathematical Problems in Engineering*, vol. 2015, Article ID 869627, 7 pages, 2015.

[19] W. Pan, Q. L. Zhong, X. D. Fu, and Z. Y. Yu, "Design and implement of workflow engine based on spring," *Journal of Northeast Normal University (Natural Science Edition)*, vol. 3, 2007.

[20] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu, and W. Zhou, "A comparative study of containers and virtual machines in big data environment," in *Proceedings of the IEEE 11th International Conference on Cloud Computing (CLOUD)*, pp. 178–185, IEEE, San Francisco, CA, USA, 2018, July.

[21] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in cloud computing: state of the art and research challenges," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 430–447, 2017.

[22] J. Zhao, X. Zhu, and L. Wang, "Study on scheme of outbound railway container organization in rail-water intermodal transportation," *Sustainability*, vol. 12, no. 4, 2020.

[23] K. A. Kuzmicz and E. Pesch, "Approaches to empty container repositioning problems in the context of Eurasian intermodal transportation," *Omega*, vol. 85, pp. 194–213, 2019.

[24] L. Knapcíková and P. Kaščák, "Sustainable multimodal and combined transport in the European Union," *Acta Logistica*, vol. 6, no. 4, pp. 165–170, 2019.

[25] J. Ližbetin, "Methodology for determining the location of intermodal transport terminals for the development of sustainable transport systems: a case study from Slovakia," *Sustainability*, vol. 11, no. 5, 2019.

[26] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 9241112, 20 pages
https://doi.org/10.1155/2022/9241112

Research Article

Advanced Phasmatodea Population Evolution Algorithm for Capacitated Vehicle Routing Problem

Jiawen Zhuang,[1] **Shu-Chuan Chu**,[2] **Chia-Cheng Hu**,[3] **Lyuchao Liao**,[1] **and Jeng-Shyang Pan**[1,2,4]

[1]*School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China*
[2]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[3]*College of Artificial Intelligence, Yango University, Fuzhou 350015, China*
[4]*Department of Information Management, Chaoyang University of Technology, Taichung 413310, Taiwan*

Correspondence should be addressed to Jeng-Shyang Pan; jengshyangpan@gmail.com

Received 30 June 2021; Accepted 24 February 2022; Published 20 March 2022

Academic Editor: Chi-Hua Chen

Capacitated Vehicle Routing Problem (CVRP) is difficult to solve by the traditional precise methods in the transportation area. The metaheuristic algorithm is often used to solve CVRP and can obtain approximate optimal solutions. Phasmatodea population evolution algorithm (PPE) is a recently proposed metaheuristic algorithm. Given the shortcomings of PPE, such as its low convergence precision, its nature to fall into local optima easily, and it being time-consuming, we propose an advanced Phasmatodea population evolution algorithm (APPE). In APPE, we delete competition, delete conditional acceptance and correspondingevolutionary trend update, and add jump mechanism, history-based searching, and population closing moving. Deleting competition and conditional acceptance and correspondingevolutionary trend update can shorten PPE running time. Adding a jump mechanism makes PPE more likely to jump out of the local optimum. Adding history-based searching and population closing moving improves PPE's convergence accuracy. Then, we test APPE by CEC2013. We compare the proposed APPE with differential evolution (DE), sparrow search algorithm (SSA), Harris Hawk optimization (HHO), and PPE. Experiment results show that APPE has higher convergence accuracy and shorter running time. Finally, APPE also is applied to solve CVRP. From the test results of the instances, APPE is more suitable to solve CVRP.

1. Introduction

The population-based algorithm is a kind of metaheuristic algorithm, and it is widely used in transportation [1, 2]. Particle swarm optimization (PSO) [3, 4], equilibrium optimizer [5, 6], flower pollination algorithm (FPA) [7–9], fish migration optimization (FMO) [10], and quasiaffine transformation evolutionary (QUATRE) [11, 12] are some popular population-based algorithms. PPE, as a novel swarm intelligence algorithm, is proposed by Song [13], and the enlightenment of PPE is from the stick insect population's evolution process.

As a novel population-based algorithm, PPE was introduced in 2020. It has some merits, e.g., the principle of the algorithm is simple and easy to implement. However, there are also some disadvantages, e.g., it is time-consuming, it has low precision, and it falls into the local solution easily. Therefore, the improvement of PPE is a challenging and meaningful study.

VRP was proposed by Dantzig and Ramser in 1959 and has been developed for many years [14], and as a branch of VRP, the related research of CVRP was very popular in transportation. CVRP refers to some customers with known demands served by some vehicles with certain capacity limits.

CVRP is an important issue. On the one hand, it is a practical problem because many real-world problems can be abstracted into CVRP. For example, in a restaurant, a waiter needs to serve multiple tables to meet customers' order needs. When only one waiter is considered, the problem is

abstracted into CVRP. Another example is that the heart of the human body supplies blood to other organs, and the blood vessels are regarded as pathways. After simplification and hypothesis, this process can be modeled as a CVRP. On the other hand, CVRP is also a challenging scientific issue. VRP is an NP-hard problem [15, 16] that is difficult to a use precise algorithm to get the optimal solution in finite time.

Many scholars have studied CVRP. In 2006, Wu et al. used a new real number encoding method of PSO for solving VRP [17]. Chen et al. proposed a novel hybrid algorithm for CVRP, and in the hybrid algorithm, discrete PSO searches for optimal results and simulated annealing are used to jump out of the local optimum [18]. In 2009, Ai and Kachitvichyanukul proposed two solution representation methods, namely SR-1 and SR-2, for CVRP [19]. In 2015, Zhang and Lee proposed a routing-directed artificial bee colony algorithm to solve CVRP [20]. In 2019, Altabeeb et al. put forward an improved hybrid firefly algorithm for CVRP, and it was tested by 82 instances [21]. In 2020, Khairy et al. put forward the enhanced group teaching optimization algorithm to solve CVRP, and it was tested by 14 instances with promising time results [22]. In 2021, Fu et al. proposed the parallel equilibrium optimizer algorithm to solve CVRP [23].

In this paper, the proposed APPE deletes competition, deletes conditional acceptance and corresponding evolutionary trend update, combines a novel jump mechanism, and adds a history-based searching for population's position update and the population closing moving method for early phase searching. Then, the algorithms are tested by CEC2013 [24]. We also apply APPE to solve CVRP.

The following is the remaining of this paper. The CVRP model is described in section 2. Section 3 introduces PPE. APPE is described in section 4. In section 5, the experiments of CEC2013 functions and CVRP are described. In section 6, a conclusion is given.

2. CVRP

The objective of CVRP is to minimize the sum of distance. The path taken by each vehicle satisfies the capacity constraint. There is only one warehouse or depot. The model of CVRP is as follows:

$$\min f(X) = \sum_{k=1}^{K_{\text{vehicle}}} \sum_{i=0}^{N_c} \sum_{j=0}^{N_c} c_{ij}^k x_{ij}^k. \tag{1}$$

s. t.

$$x_{ij}^k = \begin{cases} 1 & \text{if vehicle } k \text{ go from customer } i \text{ to } j, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$
$$i = 1, 2, \ldots, N_c;\ j = 1, 2, \ldots, N_c;\ k = 1, 2, \ldots, K_{\text{vehicle}},$$

$$y_i^k = \begin{cases} 1 & \text{if vehicle } k \text{ serves customer } i, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$
$$i = 1, 2, \ldots, N_c; k = 1, 2, \ldots, K_{\text{vehicle}},$$

$$\sum_{i=1}^{N} d_i y_i^k \le Q_k, k = 1, 2, \ldots, K_{\text{vehicle}}, \tag{4}$$

$$\sum_{k=1}^{K} y_t^k = 1, i = 1, 2, \ldots, N_c, \tag{5}$$

$$\sum_{i=1}^{N} x_{ij}^k = y_j^k, i = 1, 2, \ldots, N_c;\ k = 1, 2, \ldots, K_{\text{vehicle}}, \tag{6}$$

$$\sum_{j=1}^{N} x_{ij}^k = y_i^k, i = 1, 2, \ldots, N_c;\ k = 1, 2, \ldots, K_{\text{vehicle}}, \tag{7}$$

$$\sum_{i=0}^{N} x_{it}^k = \sum_{j=0}^{N} x_{tj}^k, t = 1, 2, \ldots, N_c;\ k = 1, 2, \ldots, K_{\text{vehicle}}, \tag{8}$$

$$\sum_{i,j \in S \times S} x_{ij}^k \le |S| - 1,\ S \subset \{1, 2, \ldots, N\},\ S \ne \Phi, \quad k = 1, 2, \ldots, K_{\text{vehicle}}, \tag{9}$$

where N_c is the customer number, K_{vehicle} is the vehicle number, c_{ij}^k is the distance from the ith customer to the jth customer by the kth vehicle, and d_i is the ith customer's demand. The kth vehicle's capacity is Q_k.

Formula (1) is the objective function. Formulae (2) and (3) describe the decision variables. Formula (4) shows the vehicle's capacity constraint. Equation (5) guarantees that every customer is served only once. Equations (6) and (7) ensure that one vehicle serves one custom. Equation (8) ensures the continuity of the route so that every vehicle coming in from the customer point would go out from that point, as well as back to the depot. Formula (9) is to eliminate the subloop.

3. PPE

PPE is inspired by the evolution process of the stick insect population. PPE initializes Np solutions randomly like other population-based algorithms. Every solution x has two properties, the first property is population number p, and the second property is population growth rate a. ev reflects the current evolution trend.

Then, calculate the fitness value, and find the global optimum, which is denoted as g best. The first k best solutions are stored in Ho, and k is equal to $\log(Np) + 1$.

Next, in iteration, the new position is the sum of old position and evolution trend, which is as follows:

$$x^{t+1} = x^t + ev, \tag{10}$$

where t is the current generation, and x^t means the current solution's position. Then, calculate new solutions' fitness and update the global optimum $gbest$ and Ho.

When the new solution's fitness is better, the current solution is absolutely replaced with it. Growth rate a, population quantity p, and population evolution trend ev are updated as follows:

$$a^{t+1} = a^t\left(1 + \frac{f\left(x^t\right) - f\left(x^{t+1}\right)}{f\left(x^{t+1}\right)}\right), \tag{11}$$

$$p^{t+1} = a^{t+1} * p^t\left(1 - p^t\right), \tag{12}$$

$$\begin{aligned} ev^{t+1} = {} & \alpha * \left(1 - p^{t+1}\right)\left(s\left(Ho, x^t\right) - x^t\right) \\ & + p^{t+1} * ev^t + \beta * p^{t+1} * m. \end{aligned} \tag{13}$$

In equation (13), the population evolution trend ev consists of three parts. In the first part, $s(Ho, x^t)$ means the nearest solution to x^t in Ho. This part reflects similar evolution. The second part preserves the inertia of evolutionary trends. The final part is the mutation.

When the new solution's fitness is worse, the current solution is conditionally updated by the new solution. If a generated random number in [0, 1] is less than the population number p, namely acceptance probability, the worse solution is accepted, and growth rate a and population quantity p are conditionally updated as equations (11) and (12). The evolution trend ev will change, irrespective of whether conditional acceptance probabilities are met, as follows:

$$ev^{t+1} = \left(s\left(Ho, x^t\right) - x^t\right) * \text{rand} + st^{t+1} * rdn\ , \tag{14}$$

$$st^{t+1} = \left(\frac{1}{t}\right)\frac{\left|f\left(x^{t+1}\right) - f_{\min}^{t+1}\right|}{f_{\max}^{t+1} - f_{\min}^{t+1}} * c^{t+1}, \tag{15}$$

$$c^{t+1} = 0.99 * c^t, \tag{16}$$

where st^{t+1} controls the exploration range. c initializes to 2, and $c$is updated by using (16), when the new solution of the algorithm is a worse solution.

Competition also affects population evolution trend. When two solutions' distance is less than G, competition will occur. Population quantity p and population evolution trend ev are updated in competition as follows:

$$p_i = p_i + a_i p_i\left(\frac{1 - p_i - p_j f\left(x_j\right)}{f\left(x_i\right)}\right), \tag{17}$$

$$ev^{t+1} = ev^{t+1} + \frac{f\left(x_j\right) - f\left(x_i\right)}{f\left(x_j\right)}\left(x_j - x_i\right), \tag{18}$$

where x_j is a random selected solution from $Np - 1$ solution. The distance from x_i is less than that from G. PPE's pseudocode is shown in Figure 1.

4. Advanced PPE

PPE also has shortcomings, such as low convergence precision, high consumption of time, and it falls into local optimum easily. Given the shortcomings of PPE, we propose APPE, which deletes competition, deletes conditional acceptance and correspondingevolutionary trend update, and adds jump mechanism, history-based searching, and population closing moving.

4.1. Without Competition. The competition mechanism exists in many algorithms. The imperialist competitive algorithm is based on imperial competition, in which weaker empires collapse and stronger empires take over colonies [25]. In the PPE, the competition mechanism is also considered. When the distance between the two solutions is too close, the population quantity p and population evolution trend ev will be updated. However, when the algorithm converges to the global or local optimal solution, this competition mechanism will affect its convergence tendency. At the same time, the calculation of particle distance takes time. Therefore, we remove the competition mechanism, thus effectively reducing the algorithm's running time. It is equivalent to deleting the $di\ st(xj, xi) < G$ part of the PPE pseudo-code.

4.2. Without Conditional Acceptance and CorrespondingEvolutionary Trend Update. Some algorithms adopt the method of conditionally accepting a worse solution to improve the algorithm diversity, such as simulated annealing [26, 27]. Similarly, PPE will adopt a worse solution with probability, increasing the algorithm diversity and also increasing the algorithm running time. Therefore, we delete the conditional acceptance and corresponding evolutionary trend update of the PPE to save time consumption and maintain a certain convergence trend. It is equivalent to deleting the $f(newx) > f(x)$ part of the PPE pseudo-code.

4.3. Jump Mechanism. We introduce a kind of jump mechanism, increasing the algorithm's probability of jumping out of local optimum. When the algorithm enters the late iteration, the optimal solution will either keep the INV generation unchanged or the standard deviation of the optimal solution in the INV generation will be less than the threshold value $INVGate$, and we let the solution jump by the following formula:

$$x_n^{t+1} = g\,\text{best} + \tan\left(\pi\left(r_0 - 0.5\right)\right) * \left(g\,\text{best} - x_n^t\right), \tag{19}$$

where x_n^{t+1} is a $t + 1$ generation solution modified by the jump mechanism to replace the nth solution. x_n^t is a selected solution from population. g best is current generation's optimal solution. r_0 is the random number, and it obeys uniform distribution from 0 to 1. In this paper, we use the jump mechanism to modify Jump Num = 5 solutions, i.e., the worst solutions from the second to the sixth. When the probability is greater than r_{Jump} and the current population is not less than 8, we delete the worst solution. Moreover, when the number of Ho is more than the current population number, the number of Ho is reset again by $k = \log(Np) + 1$, and the redundant poor solutions are deleted as well.

4.4. History-Based Searching. Most swarm intelligence algorithms use the current information to interact with the

```
Algorithm 1 : PPE

Initialize Np solutions, evolution trend ev=0 , population number p = 1/Np , population growth
  rate a = 1.1 and k = log (Np) + 1
Calculate fitness and find global best solution gbest and initialize the Ho
While not meet termination do
    Update x to newx use (10)
    Calculate new solution's fitness, update gbest and Ho
    for i = 1 to Np do

        if f (newx) ≤ f (x) then

            Accept better solution
            Update a_i , p_i and ev_i use (11), (12), (13)

        if f (newx) > f (x) then

            if rd < pi then
                Accept worse solution
                Update a_i and p_i use (11), (12)
            Update ev_i use (14), (15) and (16)

        Randomly choose a solution x_j, j ≠ i

        if dist (xj ,xi) < G then
            Update p_i use (17), updated ev_i use (18);
        if pi ≤ 0 or ai ≤ 0 or ai > 4 then
            Eliminate x and reinitialize
```

Figure 1: PPE's pseudocode.

information of the last generation, and the information before the last generation is lost, however, this information will also affect the convergence performance of the algorithm. Therefore, we design a history-based container HA, whose capacity is $HAnum$.

If the container is not full, the optimal new solution generated in each round adds to the container, which is the best particle of x^{t+1} generated by equation (10). Otherwise, the global optimal solution g best of the previous generation is used to replace any solution in the container. Meanwhile, in the second half of the iteration, if the random number generated is less than 0.5, the worst new solution generated in each round is used to replace any solution in the container, which is the worst particle of x^{t+1} generated by equation (10).

Based on the above container, we design a history-based searching method, which is used to update the population. The history-based searching formula is as follows:

$$x_{HA,i}^{t+1} = x_{HA\text{best}} + r_1\left(x_i^t - x_{H\text{Arandom}}\right) + r_2\left(x_p^t - x_q^t\right), \quad (20)$$

where x_{HA}^{t+1} is a new solution generated by history-based searching method, $x_{HA\,\text{best}}$ is a solution randomly selected from the top 5 better solutions of HA. $x_{H\,\text{Arandom}}$ is randomly selected from HA. x_i^t means the ith solution of the current iteration, and x_p^t and x_p^t are the randomly selected solutions from the population that are different from x_i^t. r_1 is an indirect random number, and $r_1 = 0.1\tan(\pi(r_3 - 0.5))$. r_2 and r_3 are the direct random numbers, and they obey uniform distribution from 0 to 1.

In this paper, the population position update of APPE combines PPE's update with history-based searching, and the detailed equation is as follows:

$$\text{new } x_i^{t+1} = \begin{cases} x_i^t + ev_i^t, & \text{if t is less than H Anum or,} \\ & \text{rand is less than} < 0.5, \\ x_{HA\,\text{best}} + r_1\left(x_i^t - x_{H\,\text{Arandom}}\right), & \\ + r_2\left(x_p^t - x_q^t\right), & \text{otherwise,} \end{cases} \quad (21)$$

where t is the current generation. When a random number $r$and is less than 0.5 or the current generation is no more than $HAnum$, the population position is updated by equation (10) or equation (20).

4.5. Population Closing Moving. Swarm intelligence algorithm in the early stage is mainly exploration. Here, we use HA, and in the first half of the iteration cycle of the algorithm, when the random number is greater than 50%, the whole PPE population is moving toward the HA solution to produce new solutions. The moving formula is as follows:

$$\mathrm{PCM}x^{t+1} = x^t + \mathrm{step} * \left(x^t - x_{HA}^{t+1}\right), \tag{22}$$

$$\mathrm{step} = \frac{2 * rdn1}{rdn2 * \mathrm{rand}}, \tag{23}$$

where x^t and x_{HA}^{t+1} are the whole population, $r\ dn\ 1$ and $r\ dn\ 2$ are random number matrices that follow standard normal distribution, and $ran\ d$ is the random number matrix that follows uniform distribution from 0 to 1. The number of rows in x^t, x_{HA}^{t+1}, $r\ dn\ 1$, $r\ dn\ 2$, and $r$and matrix is the population size, and the number of columns is the dimension. The $*$ symbol is the multiplication of the corresponding positional elements of the matrix and is neither matrix multiplication nor convolution. Notice that $2 * r\ dn\ 1$ means that every element of the matrix $r\ dn\ 1$ is multiplied with 2. Then, we compare the new solution by equation (21)with thenew solutiongenerated by moving method (18) and retain the excellent solution as the new solution.

4.6. APPE. In this paper, we combine the above methods to form APPE. The following are the detailed steps of APPE.

(1) Initialization: Np stick insect populations x^t are initialized, and p, a, and ev of each stick insect are initialized to $1/Np$, 1.1, and 0, respectively. Initialize the parameter k of Ho, and $k = \log(Np) + 1$. Initialize parameter $INV = 4$ of jump mechanism and threshold value INV Gate $= 10^{-6}$. Initialize parameter $HAnum$ of history-based archive HA and $HAnum = Np$. Initialize current iteration $t = 1$ and the maximum iteration MAXGENS. Calculate fitness $f(x^t)$. Update the global best solution $g$best and global best value $g$bestval. If $t = 1$, initialize Ho with the current best k solutions. The global best solution $g$best is put into HA.

(2) Jump mechanism: If the global best solution $g$best has kept the INV generation unchanged and the algorithm enters the late iteration or the standard deviation of the optimal solution in the INV generation is less than the threshold value INVGate, then the jump mechanism is applied. Using (19) to modify worse solutions from the second to the sixth. If a uniform random number is larger than probability r_{Jump} and the current population number is not less than 8, then delete the worst solution of PPE population and HA solutions. Moreover, when the number of Ho is more than the current population number, the number of Ho is reset again by $k = \log(Np) + 1$, and the redundant poor solutions are deleted.

(3) Population position updating: In our algorithm, equation (21) is used for position updating. New solution is new x_i^{t+1}. Calculate fitness $f(\mathrm{new}\, x^{t+1})$.

(4) Population closing moving: In the first half of the iteration cycle of the algorithm, when the random number is greater than 0.5, the whole PPE population is moving toward the HA solution, and new solutions PCM_x^{t+1} are generated by equation (22). Calculate fitness $f(\mathrm{PCM}_x^{t+1})$. Then, compare the new solution generated by equation (22) with the new solution generated by equation (21), the good one is reserved as current iteration's new solution.

(5) HA and Ho update: If HA is not full, HA is filled one-by-one with optimal new solution of each round. If $t > HAnum$, the $g$best of each round replaces the random one solution of HA. When the number of iterations is more than half, the algorithm has a 50% chance of randomly replacing one of the HA solutions with the worst new solution per round. Update the global best solution $g$best and the global best value $g$bestval. If $t > 1$, the current best k solutions are comparing with the corresponding solution of Ho, the good ones replaces the solution in Ho.

(6) Absolutely accept: When the new solution's fitness is better than that of the old one, accept it, and update p, a, and ev of each stick insect by equations (11)–(13).

(7) Parameter border check: If $p_i \le 0$, $a_i \le 0$, or $a_i > 4$, reinitialize the corresponding stick insect.

(8) Termination: Steps 2 to 7 are repeated until reaching the maximum generation. Finally, record the best fitness $gbestval$ and the best solution $g$best.

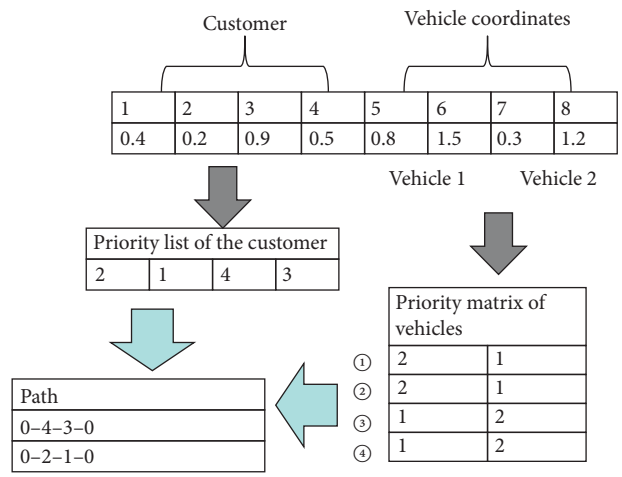


Figure 2: SR-1 method example.

4.7. Setting for CVRP. We adopt the SR-1 method for solution representation, which means that the dimension of the PPE's particle is $n + 2\,\mathrm{m}$ [19]. The first n dimensions of the solution are related to the customer, and each dimension represents the corresponding customer. The smaller the value of this dimension, the higher the priority of the customer. Thus, the priority list of the customer can be obtained. The last dimension of $2\,\mathrm{m}$ represents the coordinates of m vehicles. According to the distance of coordinates, the priority matrix of vehicles for each customer can be obtained. Arrange customer points to vehicle route one byone according to customer priority order. For a customer point, which vehiclepath it is arranged into is determined by

Table 1: Benchmark functions of CEC2013 (Test APPE's performance).

No.	Type	Optimum	No.	Type	Optimum
F1		−1400	F15		100
F2		−1300	F16		200
F3	Unimodal	−1200	F17	BasicMultimodal	300
F4		−1100	F18		400
F5		−1000	F19		500
F6		−900	F20		600
F7		−800	F21		700
F8		−700	F22		800
F9		−600	F23		900
F10	BasicMultimodal	−500	F24	Composition	1000
F11		−400	F25		1100
F12		−300	F26		1200
F13		−200	F27		1300
F14		−100	F28		1400

Table 2: Comparison of the best result of 51 runs on 10D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	739.79074	0	1.548022	1.5277E-06	0
F2	3699523.589	16892.6651	675269.84	16924.6562	68.614074
F3	588747385.1	13.3876934	689596211	74894.9703	0.000959781
F4	7128.473	988.32578	7833.35511	121.0451	68.06974
F5	50.3104	0	39.12572	8.5511E-05	1.05729E-11
F6	68.95087	0.0109923	4.216561	8.1355E-06	1.07889E-09
F7	55.7164	15.0799	47.66596	7.27598	0.44489
F8	20.1983	20.1385	20.1618	20.1326	20.2161
F9	7.2138	2.78402	5.1737	1.90882	0.822382
F10	78.52804	0.243969	13.60727	0.135769	0.0270174
F11	46.2114	3.97984	29.0232	5.1115E-06	2.69438E-11
F12	55.7539	15.9193	24.9419	14.9245	4.9748
F13	61.5672	23.5856	37.8746	16.0905	8.16871
F14	761.00596	35.1763	369.54088	137.2404	12.39519
F15	1162.7414	515.11843	208.5453	225.1287	458.76209
F16	0.780144	0.190384	0.260976	0.10284	0.679173
F17	89.55631	19.0217	41.6638	10.4437	5.15832
F18	91.67574	31.03869	34.0257	12.8986	19.6821
F19	11.1484	0.351582	3.319064	0.0606383	0.348941
F20	3.60832	3.11728	3.22622	1.83506	1.2217
F21	454.7408	200	306.5969	100.0283	400.1939
F22	1287.114	178.2702	291.18665	15.40419	14.11783
F23	1497.5974	765.85882	986.55529	73.788553	271.18111
F24	184.698	132.6267	124.3565	138.2729	105.7783
F25	173.5373	208.2578	219.6212	132.3753	109.5371
F26	168.4389	119.8991	132.0062	119.9186	105.9698
F27	560.8726	312.6709	415.0833	307.2788	301.8725
F28	476.735	100	734.1271	100.0221	100
W/D/L	0/0/28	3/0/25	1/0/27	6/0/22	20/0/8

the vehicle priority matrix.According to the priority order of vehicles, arrange a customer point into thevehicle path with higher priority. If the current vehicle path meets thecapacity constraint, arrange the customer in this vehicle path. If the currentvehicle path exceeds the capacity constraint, place the customer in the lowerpriority vehicle path. Then, the corresponding path is formed. The example of SR-1 is as shown in Figure 2.

We use some local searching methods to optimize the routes. These methods are divided into inter-route optimization and intra-route optimization. Inter-route optimization mainly consists of three-point communication and deleting-adding communication, and the search time is the sum of the instance's points and vehicles. Three-point communication refers to the random selection of three paths in the solution. A point of each path is randomly selected, and the points of the path are exchanged in the order of 3, 1, and 2. If the newly generated three paths are better than the original three paths, then the original path will be replaced [28]. Deleting-adding communication refers to the random

Table 3: Comparison of the mean result of 51 runs on 10D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	1416.5787	1.605E-13	811.9416	9.9958E-06	2.675E-14
F2	10084933.05	197537.945	6878233.51	78209.7076	6115.1032
F3	3118637483	152909331	7255467596	116898667	944921.1013
F4	12971.2353	7083.6651	15628.6068	641.8292	702.1065
F5	83.7238	1.226E-13	398.3362	0.00037539	4.16834E-10
F6	116.0168	12.8841	107.4564	18.6264	7.63911
F7	78.7358	84.1336	105.41	37.9462	16.6702
F8	20.3327	20.3472	20.3168	20.3303	20.4067
F9	8.94373	7.22868	8.71324	4.84641	3.76595
F10	181.5071	2.21159	178.7806	0.627955	0.206199
F11	69.3683	16.8379	73.7286	0.254539	3.49337
F12	79.6579	59.2814	80.4332	38.9032	17.5971
F13	80.0367	62.5323	97.0562	36.2642	29.1881
F14	1314.4366	371.4428	1017.8572	354.5957	286.6114
F15	1529.7766	1120.4131	872.7005	715.3729	1080.2465
F16	1.13441	0.661732	0.747968	0.357594	1.15523
F17	125.3906	84.2239	96.2605	11.3378	19.3253
F18	129.3592	139.6976	95.3178	31.9006	40.797
F19	35.1423	1.96173	150.0135	0.806923	0.900422
F20	3.97465	3.73039	3.97293	3.01931	2.82778
F21	507.7433	396.2685	413.4744	388.4232	400.1939
F22	1603.7803	544.8418	1458.3753	362.7106	296.2929
F23	1842.0891	1518.7686	1649.8481	1124.1341	1095.8663
F24	223.9939	221.1283	226.6401	216.8546	198.7732
F25	216.4075	222.4582	228.3634	212.0031	192.7499
F26	196.4968	197.3708	232.3701	161.1619	154.8035
F27	629.3467	562.4606	638.4348	393.6886	387.5239
F28	746.0777	699.0044	912.4083	578.1929	344.4934
W/D/L	0/0/28	1/0/27	1/0/27	8/0/20	18/0/10

Table 4: Comparison of the standard deviation result of 51 runs on 10D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	336.29231	1.1409E-13	840.7906	6.8679E-06	7.3986E-14
F2	3591008.967	126630.817	4663751.77	43253.9733	6050.4934
F3	1017194416	357131948	4644005056	183252405	3551542.55
F4	3315.33688	3588.503	2250.98706	404.5624	745.7658
F5	17.8996	5.4962E-14	321.9183	0.00018596	4.76424E-10
F6	22.96358	18.2449	56.81199	27.8417	8.28299
F7	12.4518	39.8206	38.77687	19.4452	13.3872
F8	0.0648398	0.0660583	0.0803062	0.084776	0.0790714
F9	0.489928	1.4337	1.17197	1.4601	1.48489
F10	49.44543	1.39021	120.7778	0.364935	0.115752
F11	8.12773	9.77683	28.7959	0.684481	2.17383
F12	9.1472	27.3009	31.8621	15.4218	7.28016
F13	8.76245	22.1908	28.2487	11.3712	10.3039
F14	162.82672	204.7426	238.51942	141.844	190.2989
F15	136.11119	289.33376	279.989	294.5977	301.37034
F16	0.178596	0.307342	0.226042	0.185098	0.212344
F17	13.70235	45.1711	26.9561	0.595998	7.21938
F18	15.4126	28.03248	24.9926	8.60915	7.23762
F19	19.6514	1.16874	239.7625	0.328438	0.509409
F20	0.156375	0.26237	0.341358	0.415301	0.609363
F21	25.66769	28.03275	24.7686	58.84134	3.21555E-14
F22	139.6671	261.2185	414.21438	178.3056	182.963
F23	142.08612	346.663	349.63326	388.70003	494.49528
F24	9.186187	13.39515	17.40012	16.21229	32.52391
F25	14.01808	4.603786	3.639165	15.41364	32.96771
F26	7.597635	56.24498	64.94274	34.55174	41.58459
F27	22.37158	85.9428	89.94543	33.9609	41.40027
F28	130.9133	265.6456	89.81761	225.5822	191.5919
W/D/L	12/0/16	1/0/27	2/0/26	5/0/23	8/0/20

Table 5: Comparison of the average running time result of 51 runs on 10D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	0.727436531	0.8287859	1.15685042	7.18115574	0.671487292
F2	0.791050904	0.924853635	1.3439542	7.20013287	0.70339359
F3	0.774887343	0.907082424	1.29866165	7.21305981	0.790544096
F4	0.763615047	0.903257461	1.30124119	7.14854095	0.386089429
F5	0.764020171	0.887531624	1.26940391	7.26179316	0.686430051
F6	0.726979804	0.844920447	1.16895654	7.2806207	0.670751492
F7	0.894148461	1.040195514	1.60901454	7.41189731	0.803711176
F8	0.855579947	1.018726375	1.54419505	7.22193355	0.322570457
F9	2.219833976	2.641704835	4.92032594	8.67749507	2.310656547
F10	0.773372886	0.891785363	1.28616139	7.27650189	0.668267453
F11	0.816651865	0.962051961	1.4769969	7.27590196	0.746390108
F12	0.843791384	0.99691912	1.51443125	7.49711857	0.609939676
F13	0.83997619	0.98404201	1.48591037	7.3828366	0.510813414
F14	0.822789535	0.943898739	1.47654896	7.28391574	0.582743498
F15	0.829717169	0.957212175	1.48187512	7.38718168	0.461557804
F16	1.83420029	2.180153114	4.0038432	8.25544575	1.247958082
F17	0.781633394	0.907479276	1.36932876	7.24234632	0.577252965
F18	0.79514379	0.932933855	1.42243596	7.20300338	0.388129488
F19	0.743986608	0.862446631	1.27623414	7.29507605	0.715902753
F20	0.801560351	0.916730906	1.4320129	7.30689116	0.494742363
F21	1.065321725	1.263061847	1.98997494	7.57730783	0.883767925
F22	1.159216551	1.320134182	2.28246203	7.58581411	0.931372729
F23	1.168561278	1.348461751	2.30642576	7.61082738	0.817745671
F24	2.571608424	3.024779602	5.6708838	8.99084936	2.541635441
F25	2.591145233	3.013362006	5.71097513	8.91483909	2.597728576
F26	2.784730184	3.323031224	6.20988149	9.09719149	2.571828318
F27	2.745837339	3.261529955	6.11941155	9.10913397	2.372727906
F28	1.288557314	1.523347316	2.596839	7.62935641	1.087740337
W/D/L	3/0/25	0/0/28	0/0/28	0/0/28	25/0/3

Table 6: Comparison of the best result of 51 runs on 30D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	20965.5117	2.2737E-13	3115.0987	3.4641E-05	2.2737E-13
F2	184302669.7	556561.902	42409000.72	1551822.08	36796.10533
F3	99304872220	10402522.9	24129867958	14277792	11979.52113
F4	47671.1384	17895.2805	41958.6448	436.4169	277.24218
F5	2736.1858	4.5475E-13	845.64218	0.00122154	2.01521E-09
F6	2192.6367	2.59305	371.83971	0.304544	1.69745E-05
F7	326.5177	107.2508	227.7295456	44.6206	39.1504
F8	20.8505	20.7428	20.7042	20.8119	20.7702
F9	36.6723	27.5559	34.1636	23.3817	15.4945
F10	2740.0071	0.0640834	544.29708	0.0702674	6.77581E-08
F11	475.3545	93.52546	342.582	15.122	17.9093
F12	536.0034	303.4593	288.8458	251.7349	72.63182
F13	477.1505	333.4405	499.8886	261.1692	190.6686
F14	6284.6022	1787.2956	2549.475	766.94606	431.50235
F15	6306.8434	2999.2399	3353.0779	2500.5042	5525.0534
F16	1.64039	0.267507	0.822458	0.193588	1.58162
F17	1008.295	287.815	323.721	43.6346	65.20754
F18	1102.4916	350.691	487.3008	174.6342	156.3594
F19	43694.72938	9.84876	342.97435	2.91789	2.43391
F20	14.521	12.6801	14.5097	13.8654	9.58436
F21	2801.7822	100	929.77219	100.2957	100
F22	6985.8846	1760.6213	4701.2418	592.84675	362.0814
F23	7504.1003	3830.2633	3760.2452	3520.5278	3135.9944
F24	325.3054	282.8699	307.7325	251.8537	234.3891
F25	336.8065	291.219	310.1113	308.9248	270.7536
F26	216.1884	200.0521	204.6498	200.0465	200.0016

Table 6: Continued.

Fun	DE	SSA	HHO	PPE	APPE
F27	1365.8604	936.36559	1246.5007	945.20441	746.8349
F28	3662.6068	300	3717.7686	100.27997	100
W/D/L	0/0/28	2/0/26	1/0/27	4/0/24	22/0/6

Table 7: Comparison of the mean result of 51 runs on 30D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	32193.5619	6.5537E-13	9711.9121	8.1178E-05	4.6812E-13
F2	368242452.8	2081301.22	179774792.8	3096187.31	126305.9182
F3	$7.10045E+11$	698447995	$9.77373E+13$	434470674	31758287.64
F4	75322.0536	28653.0937	52077.4646	930.919	1330.2804
F5	4522.474	1.0867E-11	2695.5244	0.00195401	5.23523E-08
F6	3552.4961	30.175	1758.4039	59.1902	6.12643
F7	747.6163	211.4113	108509.3088	94.8052	84.0757
F8	20.9546	20.922	20.9005	20.9443	20.9755
F9	39.276	33.915	39.5487	30.821	24.7972
F10	4134.0267	0.211734	1991.4149	0.44158	0.0650486
F11	579.6098	240.4308	584.3916	28.1263	33.9158
F12	628.6948	511.8694	606.0292	353.5071	183.0126
F13	624.5928	440.3657	701.2616	379.6572	266.1043
F14	7069.1658	2731.909	4328.8102	1243.0858	1814.8754
F15	7431.2237	4754.6443	4829.5591	3951.5532	6617.3184
F16	2.49294	1.25122	1.56921	0.745032	2.49028
F17	1262.8753	652.2582	666.9315	62.4704	108.3917
F18	1269.4583	702.0829	684.1028	287.4171	300.5752
F19	167474.7719	19.342	4354.8665	6.46873	5.63526
F20	14.9018	14.7589	14.8377	14.6797	12.2536
F21	3210.8721	364.2293	1623.5516	315.6636	320.9034
F22	7723.8174	3374.4772	6251.9555	1336.6024	940.461
F23	8097.5094	5851.2557	6602.7644	5180.4845	6658.3955
F24	334.942	303.6476	339.2056	286.0378	263.803
F25	353.4227	314.6729	343.4621	348.9108	297.3444
F26	241.288	363.4894	384.4514	315.0175	200.0092
F27	1428.7773	1254.8879	1429.8648	1145.826	927.6687
F28	4413.077	3213.3913	4978.1659	3248.566	690.8847
W/D/L	0/0/28	1/0/27	1/0/27	9/0/19	17/0/11

Table 8: Comparison of the standard deviation results of 51 runs on 30D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	3909.09245	2.4761E-13	3561.6764	2.2148E-05	1.5362E-13
F2	74221490.37	941488.26	89397375.81	816426.966	83074.86122
F3	$7.20796E+11$	1065594457	$3.23802E+14$	358944299	85733576.72
F4	8466.66261	4475.09793	3760.44355	309.9896	1143.5652
F5	755.38339	1.8485E-11	1543.1381	0.00050402	5.65084E-08
F6	640.64055	24.7472	1147.0894	25.2244	6.49659
F7	393.6556	106.0421	271133.1512	19.8107	26.8212
F8	0.0491254	0.0573101	0.0718902	0.0532219	0.0504813
F9	1.1558	3.41188	2.12293	2.9527	3.79688
F10	540.80016	0.114618	984.78042	0.220342	0.0490094
F11	44.57488	75.69394	105.7453	7.26775	9.97949
F12	40.49799	134.3097	119.1011	53.33428	50.99063
F13	46.1267	76.22904	86.99332	48.55205	40.97781
F14	262.46923	535.39708	819.82348	236.31161	1441.8054
F15	271.96764	730.40653	835.72922	627.96769	417.02913

Table 8: Continued.

Fun	DE	SSA	HHO	PPE	APPE
F16	0.320608	0.547221	0.43001	0.300076	0.346913
F17	107.2886	136.14	97.29982	7.74407	21.66193
F18	78.052688	126.3105	83.33826	53.92487	41.8762
F19	67584.43532	5.63479	3353.4627	2.11375	1.76896
F20	0.114222	0.529039	0.231521	0.282559	0.684561
F21	172.3487	96.53369	301.79304	72.4779	85.50782
F22	269.7388	869.43735	845.09278	295.70899	339.5195
F23	243.52231	856.05385	1075.8917	756.67546	1221.6536
F24	4.086933	9.678988	18.7233	16.02537	14.32264
F25	4.969392	11.83765	16.5024	21.20163	13.09351
F26	8.458015	66.57767	64.77045	82.44704	0.005305383
F27	27.630268	109.75615	75.82382	82.797679	89.73993
F28	269.10614	1467.6047	630.94737	773.56802	916.4915
W/D/L	11/0/17	1/0/27	0/0/28	7/0/21	9/0/19

Table 9: Comparison of the average running time result of 51 runs on 30D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	2.733809878	2.65437004	4.1709789	21.8721739	2.407070994
F2	3.238093776	3.28368366	5.52267889	22.1806176	3.242242859
F3	3.387544408	3.44985781	5.79780013	22.1986548	2.802126447
F4	3.033938118	2.98933781	5.04895915	21.8332192	1.460134582
F5	3.060770578	3.05556757	4.96295184	22.2294468	2.630876359
F6	2.912188557	2.89667259	4.5023418	22.1873498	3.894232227
F7	4.585807916	4.73127061	9.05654281	23.5809211	4.413959241
F8	3.954471206	4.07066223	7.33255408	22.9174646	1.585859245
F9	17.05689298	18.9060265	40.0398491	35.2771729	12.41038709
F10	3.334638673	3.29820511	5.4339282	22.2756434	2.716332702
F11	3.444212914	3.48815126	5.99635844	22.3046879	3.0724273
F12	3.981317933	4.03849017	7.26727993	22.886067	2.921447771
F13	3.976971841	4.00729305	7.10558566	22.9836765	1.930308529
F14	3.685780082	3.67497535	6.60276835	22.7189149	2.713648625
F15	3.830101522	3.86078174	6.65525209	22.8490806	1.75100859
F16	13.45326515	14.7837753	29.2868758	31.7040965	8.918082525
F17	3.147515716	3.21549638	5.08030391	22.1923851	2.652192106
F18	3.500081333	3.59570355	5.83726951	22.3467761	1.447078869
F19	3.011889486	3.04511699	4.66262677	21.9862949	3.609727229
F20	3.670195529	3.77716328	6.92572923	22.390139	1.774093445
F21	5.8482322	6.27261929	10.5518727	24.3638556	4.451286106
F22	6.232905657	6.51323347	12.1217653	24.7453143	5.447029629
F23	6.791029171	7.08679317	13.3236218	25.1882066	4.829456082
F24	20.09962069	22.2324157	43.2164071	37.7097732	14.63043557
F25	20.0838764	22.4150292	42.3260732	37.7248481	14.5165233
F26	21.67665217	24.595264	46.7010523	39.2097254	24.28157111
F27	21.23823786	24.2659194	46.0547195	38.9343311	15.58041748
F28	8.259479129	8.96047216	16.5444839	26.6793565	6.434003169
W/D/L	3/0/25	1/0/27	0/0/28	0/0/28	24/0/4

selection of two paths in the solution, with a point randomly selected in each path. One of the paths deletes the selected points and adds the selected points to the other path. If the new two paths are better than the original two paths, then the original path is replaced [29]. Intra-route optimization consists of path scrambling, path inversion, and path 2-point swapping. The number of executions of these methods is the square of the number of points in each path of the solution. Path scrambling refers to the random rearrangement of paths, except the starting and ending points, which are rearranged using the randperm function of Matlab. If the generated new path is better than the best solution, then it replaces the original best solution, and the next searching will still change the original path, however, the comparison object is the best solution. Path inversion means that two points of the path are selected each time, and then the paths between the two points are rearranged in the reverse order. If the new solution produced is good, then retain it as the best

Table 10: Comparison of the best result of 51 runs on 50D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	68441.6877	6.8212E-13	4421.5318	9.1774E-05	1.13687E-12
F2	796828451.2	897706.17	27754539.4	2311976.79	114810.9289
F3	$4.11102E+11$	36730944.5	45009339893	64993978.4	3218012.379
F4	103396.927	33164.3416	56242.6043	233.6512	428.55484
F5	9117.13317	2.2907E-09	991.73975	0.00229062	2.2956E-07
F6	6066.8624	29.1158	444.7514	39.7167	0.128249
F7	321.54597	96.81021	282.559833	70.85565	56.4843
F8	21.0487	21.051	20.9601	21.0257	21.0471
F9	70.0516	49.8706	65.8906	48.0806	39.9872
F10	8540.84996	0.0394587	760.25367	0.857053	3.88815E-05
F11	1120.4763	382.0615	591.099	69.93432	62.6823
F12	1157.4674	587.01751	695.9241	461.6641	204.9601
F13	1132.7369	640.2518	953.41855	566.5412	421.8479
F14	12375.2343	3470.8789	5201.6947	1011.1859	1039.3238
F15	13171.1787	6238.7238	7676.88995	5808.5115	11569.9181
F16	2.68248	0.689805	1.27295	0.546151	2.42801
F17	2686.1285	502.0316	782.54144	112.7755	155.0626
F18	2698.9651	987.4102	870.71601	451.2733	360.0513
F19	466278.7647	24.3778	912.66167	7.4999	7.79172
F20	24.4448	21.9486	23.6975	21.8148	18.6627
F21	5875.3627	836.4425	1831.4087	836.4425	100
F22	13466.9914	5826.1274	7811.75043	1936.8612	1064.7735
F23	14407.6167	8117.39148	9107.61415	7311.61553	8301.79767
F24	432.5152	371.4055	415.414	327.2399	310.6605
F25	488.8236	387.3883	412.8656	434.7963	380.0998
F26	296.6734	440.0132	469.8953	200.2743	200.0272
F27	2321.1044	1775.5901	2151.7305	1692.6655	1292.2378
F28	7644.2744	400	6451.9363	474.57092	400
W/D/L	0/0/28	3/0/25	1/0/27	7/0/21	18/0/10

Table 11: Comparison of the mean result of 51 runs on 50D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	80496.6275	1.2528E-12	9600.2818	0.00017649	3.49487E-11
F2	1170022441	2226137.48	192268997.5	4042717.04	400359.6315
F3	$4.40463E+12$	1369573907	$2.04922E+11$	392037714	69788181.12
F4	136314.9379	53168.688	69134.0056	674.8522	1286.1218
F5	15201.5539	1.821E-08	1625.4642	0.00374727	1.4891E-06
F6	7949.0109	72.5883	902.9159	86.2859	35.1058
F7	1086.8641	192.4685	11581.7115	116.5335	83.5833
F8	21.1353	21.1317	21.1175	21.1177	21.1504
F9	72.941	65.2474	73.2972	57.8761	52.102
F10	10371.6231	0.202345	1961.669	1.52074	0.0368955
F11	1327.7597	547.3071	745.3906	110.0593	93.4338
F12	1336.6887	1014.754	916.3933	570.8822	412.2794
F13	1324.4015	956.3305	1168.9616	725.9276	536.2261
F14	13453.7746	5825.0496	7675.3836	1963.8542	2825.8275
F15	14290.1024	8553.19	10356.5828	7728.0556	12892.3516
F16	3.28518	1.83319	2.27207	1.2382	3.31238
F17	3020.4296	981.1095	1024.8344	146.7249	226.2728
F18	3053.1201	1155.8584	1096.2182	593.7946	627.5738
F19	1429662.26	49.627	2624.7591	13.2586	12.7787
F20	24.8789	24.3835	24.485	24.0914	21.9659
F21	7181.9716	998.9602	3051.678	948.5251	870.4084
F22	14742.4834	7413.6214	10468.5507	3276.6393	2423.8416
F23	15334.0567	10546.0751	12655.7041	10424.3902	13606.4302
F24	464.8197	391.9212	470.8258	373.7801	342.4159
F25	502.6936	415.216	459.7346	504.9336	408.2602
F26	368.9144	470.2997	495.343	411.5417	283.8361
F27	2446.8166	2076.0709	2433.0726	2107.7286	1650.236
F28	8778.8527	5148.5057	9186.8097	5936.9657	1470.4027
W/D/L	0/0/28	2/0/26	1/0/27	7/0/21	18/0/10

Table 12: Comparison of the standard deviation result of 51 runs on 50D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	5405.61377	3.9217E-13	2679.9505	4.1786E-05	7.0273E-11
F2	170612616.4	824253.494	89445562.68	1007133.92	179412.3457
F3	$3.2854E+12$	1557568439	$4.01457E+11$	379016850	94471867.3
F4	11084.63859	11809.0884	5827.04834	184.5265	668.48665
F5	2116.93995	1.7468E-08	324.03013	0.00084021	1.5852E-06
F6	955.60891	31.5493	456.9381	32.6704	18.3843
F7	525.90396	79.70547	13589.3232	21.04326	14.1085
F8	0.0363517	0.0364229	0.0481815	0.0365593	0.042277
F9	1.36303	4.67854	3.15456	3.72436	5.71436
F10	882.747601	0.115331	607.18573	0.256366	0.024425
F11	81.157358	92.02943	70.68308	20.84648	15.7099
F12	64.758226	116.40179	97.13284	56.83782	96.05942
F13	77.218129	183.0932	94.733558	69.5969	66.06756
F14	326.008925	1029.6185	1404.319	399.01654	2150.2895
F15	393.289928	960.32527	1506.75168	942.05543	609.042346
F16	0.239473	0.61187	0.531921	0.350457	0.371027
F17	146.22569	147.6494	75.297952	14.74808	36.31978
F18	125.67353	46.142709	79.748497	71.51158	94.5146
F19	322868.2211	16.5907	1522.0331	2.10674	4.34599
F20	0.1289	0.482846	0.129266	0.631614	1.0936
F21	362.66356	142.9595	313.42278	140.9302	306.9437
F22	365.40434	1019.4687	1239.58645	838.11989	1044.5246
F23	367.202851	1354.96456	1416.92288	1245.35206	1138.12779
F24	11.07239	13.63289	38.71835	25.27126	18.62976
F25	7.704087	17.48497	35.64058	36.57497	19.04745
F26	26.48411	14.2354	11.83787	105.9412	110.0795
F27	44.650405	136.27279	144.65187	159.38326	122.3399
F28	416.1356	3120.0947	991.68671	988.11438	1868.8146
W/D/L	12/0/16	3/0/25	1/0/27	5/0/23	7/0/21

Table 13: Comparison of the average running time result of 51 runs on 50D in 28 benchmark functions.

Fun	DE	SSA	HHO	PPE	APPE
F1	5.371938416	4.57383823	7.23377461	38.1038224	4.446575594
F2	7.157403912	6.62089211	11.3540526	39.1676853	7.158186551
F3	7.839430525	7.3246534	12.5236647	39.9919087	6.214457253
F4	6.484470433	5.91526308	10.1147916	38.58679	3.307264696
F5	6.159703835	5.54299413	9.14633452	37.6149204	4.968641435
F6	6.300308008	5.60679982	9.2921294	37.6352779	6.911794831
F7	11.10656359	10.7296947	20.4603011	41.5075489	10.88083133
F8	9.318202392	8.70874237	16.4839356	39.6187509	3.878486443
F9	46.05065865	48.7456533	101.389658	75.4380672	32.79673965
F10	7.527485204	6.72485453	11.9008743	40.0126535	6.521356616
F11	7.303516457	6.65447378	11.9370057	40.183341	6.179351049
F12	9.596861269	9.09086465	17.1752837	42.2132078	6.93119691
F13	9.743597688	9.1466903	17.1017582	42.4066013	4.65911769
F14	7.889016759	7.25925715	13.828576	40.8411437	6.670441894
F15	8.824859059	8.19700595	15.6963778	41.2855432	4.436260716
F16	35.43426225	38.0182758	78.9101852	66.163459	23.73375013
F17	6.545739804	5.81661149	10.4148023	38.0207576	5.578457125
F18	8.349048814	7.82492584	14.3796084	39.1149567	3.9553397
F19	6.863519427	5.93801426	10.3868958	37.8526068	7.639078567
F20	8.741159843	7.85354537	15.0667554	38.94757	3.508262818
F21	15.61953418	16.2938313	29.2424287	45.2747967	11.12546115
F22	15.09370126	15.3303718	29.6861623	45.1094114	13.92498259
F23	17.74963977	18.3327347	35.4951246	47.3597377	11.49145802
F24	55.77218786	61.9013014	118.547038	81.9097344	39.63127379
F25	52.60461786	63.011092	116.425617	82.0330892	39.2314303
F26	57.51527773	70.9420056	132.969794	86.6588994	58.37427563
F27	56.10193225	69.4253279	129.134722	85.362875	42.25193263
F28	21.45543076	24.3204501	46.9656329	51.9634444	18.04838259
W/D/L	1/0/27	4/0/24	0/0/28	0/0/28	23/0/5

Table 14: Comparison of the best result of 10 runs in CVRP instances.

Instances	BKV	DE	SSA	PSO	PPE	APPE
A-n32-k5	784	794.4865	787.0819	787.2024	787.0819	787.0819
A-n33-k5	661	678.0007	662.2642	673.1471	662.2642	662.1101
A-n33-k6	742	746.125	742.6933	742.6933	742.6933	742.6933
A-n34-k5	778	787.6693	786.437	786.437	786.7965	786.437
A-n36-k5	799	808.5726	802.1318	802.1318	802.1318	802.1318
A-n37-k5	669	688.7647	672.7407	672.7407	672.5174	672.5174
A-n37-k6	949	960.1145	956.8075	966.0936	957.0298	957.0298
A-n38-k5	730	747.4928	738.0118	734.4416	734.4416	733.9458
A-n39-k5	822	836.0773	829.4541	829.5219	829.5219	829.5219
A-n39-k6	831	841.5609	833.2046	835.2518	835.2518	835.2518
A-n44-k7	937	955.3702	943.4791	948.8242	952.5564	943.6351
A-n45-k6	944	984.3388	959.2347	994.68528	970.79579	945.3614
A-n45-k7	1146	1186.559	1159.7315	1165.812	1170.0275	1153.0785
A-n46-k7	914	942.1947	921.7679	921.8017	918.1274	918.1274
A-n48-k7	1073	1134.015	1095.2564	1106.6835	1100.3926	1094.9122
A-n53-k7	1010	1098.815	1042.8605	1054.9787	1030.8244	1045.9452
A-n54-k7	1167	1226.547	1184.6926	1188.767	1196.2087	1182.8441
A-n55-k9	1073	1135.431	1082.8527	1082.9222	1088.2412	1078.39
A-n60-k9	1408	1434.627	1364.6654	1386.0294	1369.973	1379.077
A-n61-k9	1035	1182.316	1071.5384	1147.6217	1080.8725	1104.0151
A-n62-k8	1290	1372.683	1348.8174	1339.6425	1338.6479	1331.7414
A-n63-k10	1315	1384.93	1352.951	1363.3655	1345.4533	1323.9144
A-n63-k9	1634	1725.011	1661.5273	1701.0925	1692.2855	1639.9246
A-n64-k9	1402	1501.943	1425.8711	1446.1263	1424.9294	1433.2557
A-n65-k9	1177	1267.51	1201.895	1260.8151	1186.6736	1208.8699
A-n69-k9	1168	1243.583	1183.333	1198.7175	1190.5865	1183.0729
A-n80-k10	1764	1911.073	1828.1481	1828.8761	1831.7224	1823.0898
W/D/L		0/0/27	10/0/17	3/0/24	8/0/19	18/0/9

Table 15: Comparison of the mean result of 10 runs in CVRP instances.

Instances	BKV	DE	SSA	PSO	PPE	APPE
A-n32-k5	784	802.3796	791.6953	789.5103	794.8186	788.8223
A-n33-k5	661	684.0878	675.9522	678.5661	671.0078	665.5313
A-n33-k6	742	754.7865	747.2399	745.6267	744.6511	743.8523
A-n34-k5	778	792.6542	792.4126	792.1176	792.9502	788.4026
A-n36-k5	799	816.6452	808.0167	810.3682	811.9299	807.2603
A-n37-k5	669	700.4381	689.2895	687.5505	683.889	678.6079
A-n37-k6	949	981.1734	970.0739	974.8255	970.9061	967.2458
A-n38-k5	730	758.8868	749.0006	751.4792	741.9858	740.0858
A-n39-k5	822	847.8701	837.9498	844.841	833.7223	833.781
A-n39-k6	831	852.229	838.8618	838.0234	836.9315	836.5501
A-n44-k7	937	978.7361	962.6828	969.9482	967.2578	951.4784
A-n45-k6	944	1021.858	997.9943	1038.8207	1034.7167	999.428
A-n45-k7	1146	1200.789	1182.0618	1184.3241	1185.3844	1164.4213
A-n46-k7	914	964.4004	940.755	950.2408	941.6035	933.0219
A-n48-k7	1073	1144.862	1109.3274	1112.9579	1116.9883	1103.2999
A-n53-k7	1010	1106.072	1064.1732	1073.4099	1063.4356	1057.0531
A-n54-k7	1167	1245.638	1204.771	1228.5166	1231.2475	1202.2122
A-n55-k9	1073	1150.691	1106.4484	1113.959	1117.2283	1095.4729
A-n60-k9	1408	1443.306	1396.0582	1406.6868	1408.4423	1395.4768
A-n61-k9	1035	1206.694	1160.0295	1190.2004	1139.945	1146.3305
A-n62-k8	1290	1388.059	1360.9244	1356.5061	1363.7038	1345.357
A-n63-k10	1315	1412.946	1385.1111	1393.813	1372.8607	1348.2184
A-n63-k9	1634	1743.69	1690.8434	1735.573	1727.2144	1695.1167
A-n64-k9	1402	1518.639	1463.5227	1463.5476	1448.9618	1450.4584
A-n65-k9	1177	1306.929	1244.3624	1283.902	1257.6103	1239.5486
A-n69-k9	1168	1269.933	1208.9972	1228.6695	1216.934	1203.1344
A-n80-k10	1764	1929.862	1864.3596	1874.9804	1868.4458	1852.7933
W/D/L		0/0/27	2/0/25	0/0/27	3/0/24	22/0/5

Table 16: Comparison of the best result of 10 runs of APPE with the algorithms proposed by Korayem et al.

Instance	Optimum	KmeansFnO	KmeansFnP	KmeansFnR	APPE
A-n33-k6	742	828	754	754	742.6933
A-n34-k5	778	792	822	839	786.437
A-n36-k5	799	814	846	846	802.1318
A-n39-k6	831	868	856	837	835.2518
A-n60-k9	1354	1395	1421	1423	1373.4651
A-n62-k8	1288	1366	1329	1329	1346.9532
B-n31-k5	672	684	672	672	676.0884
B-n34-k5	788	793	817	817	790.9678
B-n39-k5	549	571	564	566	553.1565
B-n41-k6	829	845	920	877	836.7407
B-n44-k7	909	940	954	944	932.0102
B-n50-k7	741	765	783	774	745.16
B-n50-k8	1312	1348	1359	1361	1341.876
W/D/L		0/0/13	2/0/11	2/0/11	11/0/2

Table 17: Comparison of the best result of 20 runs of APPE with the algorithms in Yan et al.

Instance	CO-HS	PSO	GA	APPE
A-n32-k5	807	829	818	797.7186
A-n33-k5	669	705	674	680.4111
A-n34-k5	790	832	821	786.437
A-n39-k6	852	872	866	842.8981
W/D/L	1/0/3	0/0/4	0/0/4	3/0/1

Table 18: Comparison of the best and mean results of 10 runs of APPE with the algorithms in Zhao et al.

Instance	QDE/Best	QDE/Mean	QEA/Best	QEA/Mean	DE/Best	DE/Mean	APPE/Best	APPE/Mean
P-n76-k4	718	728	718	728	746	765	615.72618	622.3824
P-n76-k5	740	751	740	751	803	813	643.98732	664.62492
A-n32-k5	784	810	835	861	910	924	787.08189	788.26212
A-n33-k5	707	716	707	716	724	735	662.2642	664.95601
A-n33-k6	785	787	785	787	857	872	742.69326	743.77421
W/D/L	1/0/4	0/0/5	0/0/5	0/0/5	0/0/5	0/0/5	4/0/1	5/0/0

Table 19: Comparison of the mean result of 10 runs of APPE with the algorithms in Khairy et al.

Instance	GA	ACO	GTO	APPE
A-n32-k5	812.9	982.5	901.7	790.3117
A-n33-k5	674.9	787.7	746.7	667.0751
A-n34-k5	803.4	911.8	862.8	790.7029
A-n36-k5	839.3	1008	903.3	809.091
A-n39-k6	894.9	1085	975.9	835.5559
A-n46-k7	987.8	1206	1138	932.0442
A-n80-k10	1987	2369	2470	1840.5895
X-n106-k14	28897	30450	31928	27751.5143
X-n129-k18	33893	38098	41710	30742.9103
X-n143-k7	21664	22674	39979	17318.072
X-n167-k10	28347	28859	48205	22904.9479
X-n209-k16	42587	42286	70080	33887.9638
W/D/L	0/0/12	0/0/12	0/0/12	12/0/0

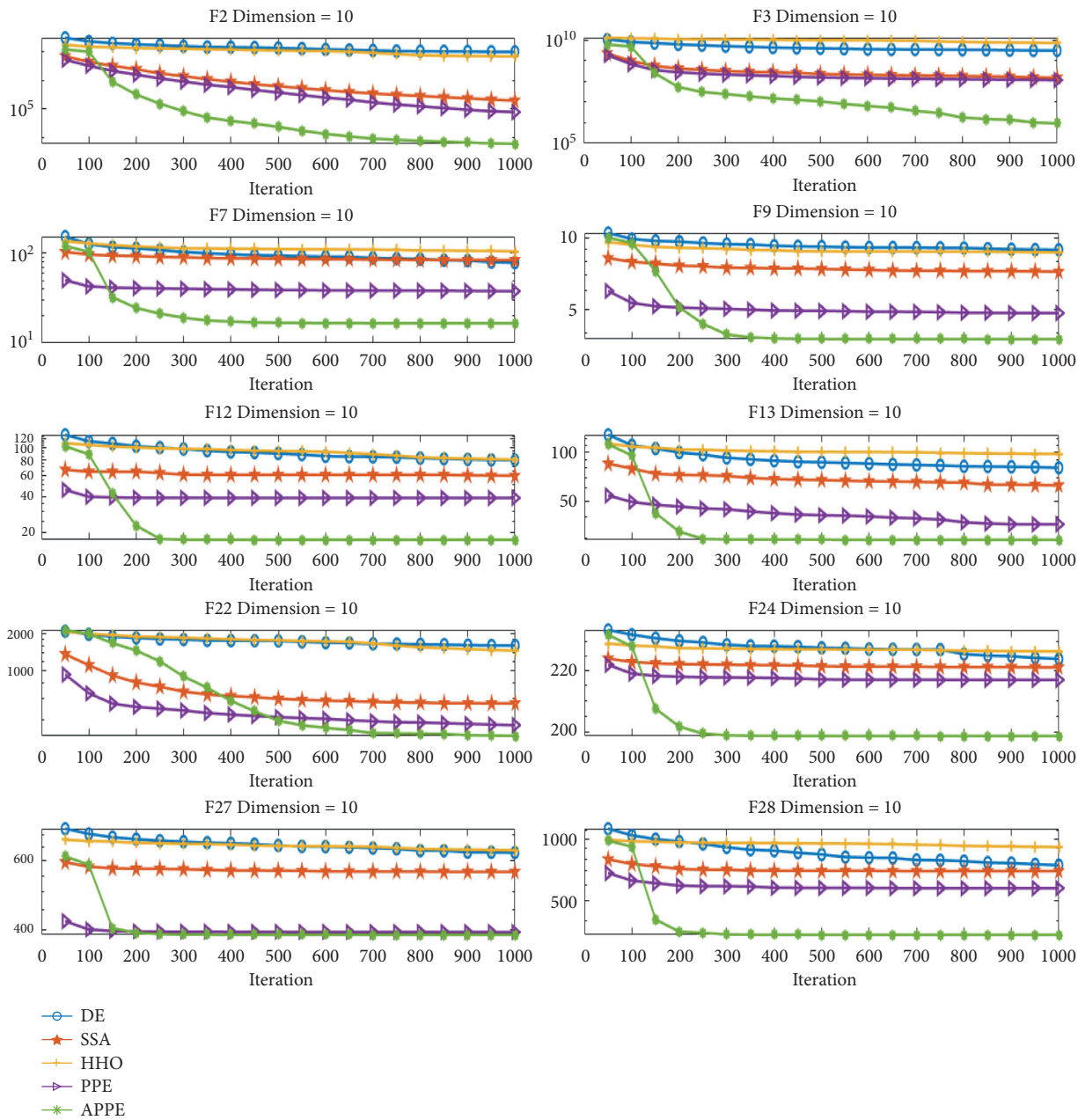


Figure 3: APPE's convergence curve with the dimension of 10.

solution for the next search [28]. Path 2-point swapping means that two points of the path are selected each time, and then the two points are exchanged. If the new solution produced is good, then retain it as the best solution for the next search [28].

5. Experiment and Application

In this section, we utilized CEC2013 to test our proposed algorithm, as shown in Table 1. APPE experiment results are shown in Tables 2–13. Then, the CVRP results are as shown in Tables 14–15. These are the results APPE compares with DE, SSA, PSO, and PPE. We also compare APPE with some existing work in Tables 6–19.

Three types of functions are included in CEC2013, as shown in Table 1. The first type is the unimodal function, which tests the exploitation ability. The second is the basic multimodal function, which tests the exploration ability. The third type is the composition function, which is composed by the above-mentioned function, representing the challenging problems. The search range is [−100, 100] for each dimension.

5.1. Experiment Setting. In this experiment, we compare APPE with DE, SSA, HHO, and PPE. Each algorithm has 100 solutions, i.e., $ps = 100$, with $Di\ m = 10, 30, 50$. Each algorithm has 51 independent runs in each benchmark, and Maxgen equal to $10000 \times \mathrm{Dim}/ps$. The parameter setting is

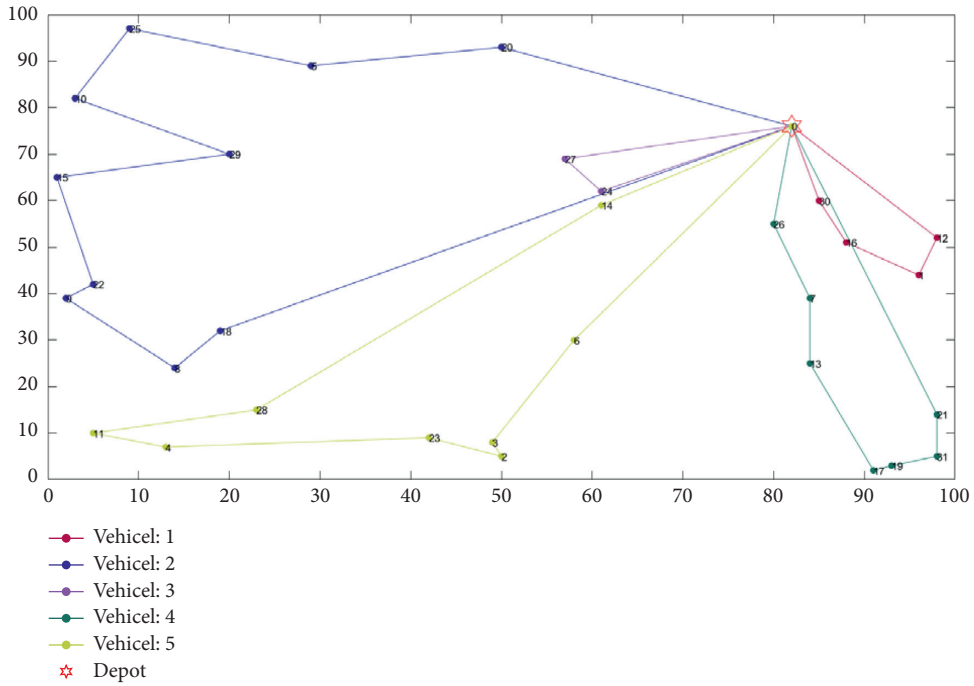


Figure 4: The best route result of the A-n32-k5 by APPE.

as follows: for DE, $F = 2$, $CR = 0.9$, and it uses DE/rand/1/bin. For HHO, $\beta = 1.5$. For SSA, the number of producers accounts for 20%, $SD = 20$, and $ST = 0.8$. For PPE, $k = \log(Np) + 1$, p, a, and ev of each stick insect are initialized to $1/Np$, 1.1, and 0, and $G = (ub - lb)((\text{Maxgen} + 1 - t)/\text{Maxgen})/10$. For APPE, $INV = 4$, $INV\text{Gate} = 10^{-6}$, JumpNum = 5, $r_{\text{Jump}} = 0.05$, $HAnum = Np = ps$, and other parameters are the same as the PPE.

The qualitative metric uses the convergence curve, and the quantitative measure comprises the best, mean, standard deviation, and average running time of the specific benchmark functions.

5.2. Experiment Test. Table 2 is the best experimental result of APPE with $Di\ m = 10$, which is the best of 51 runs. Table 3 is the mean experimental result of APPE with $Di\ m = 10$, which is the mean of 51 runs. Table 4 is the standard deviation experimental result of APPE with $Di\ m = 10$, which is the standard deviation of 51 runs. Table 5 is the time experimental result of APPE with $Di\ m = 10$, which is the average running time of each algorithm in 51 runs. Similarly, Tables 6–9 are the result of $Di\ m = 30$, and Tables 10–13 are the result of $Di\ m = 50$. We use fitness error $f - f_{\text{optimum}}$ for simplicity. We also use W/D/L to record each algorithm's win/draw/loss number in 28 benchmark functions from Tables 2–13. Under a benchmark function test, if the algorithm has the best performance (minimum fitness value), then W adds one. If the algorithm is equal to other algorithms (with the same fitness value), then D adds one. Otherwise (the algorithm's fitness value is not minimum), L adds one. Figure 3 shows the convergence curves of APPE, and the dimension is 10. APPE compares with DE, SSA, HHO, and PPE.

As shown in Tables 2, 6, 10, APPE's W/D/L of the best experimental result are 20/0/8, 22/0/6, 18/0/10, respectively. It indicates that APPE can search better solutions and is more likely to jump out of local solutions. From Tables 3, 7, 11, APPE's W/D/L of mean experimental result are 18/0/10, 17/0/11, 18/0/10, respectively. It means that APPE has a higher convergence accuracy. As shown in Tables 4, 8, 12, APPE's W/D/L of standard deviation experimental result are 8/0/20, 9/0/19, 7/0/21, respectively. Compared with other algorithms in these tables, it can be seen that APPE has moderate convergence stability. From Tables 5, 9, 13, APPE's W/D/L of time experimental result are 25/0/3, 24/0/4, 23/0/5, respectively. It shows that the running time of APPE is relatively short compared with other algorithms. Therefore, the convergence precision and running time of APPE are quite well.

As shown in Figure 3, the convergence accuracy of APPE at the beginning of iteration is at a general level. In the middle of iteration, many algorithms are in a stable convergence state, which is similar to falling into the local optimum, while APPE still continues to search at this time, which has a certain exploration ability and can jump out of the local optimum. In the latter part of iteration, it enters the stable convergence stage like many algorithms. In Figure 3, APPE's convergence curve is basically at the bottom, i.e., the convergence precision is the best. In addition, APPE convergence curve obviously has a steep slope of decline, i.e., APPE can get a better solution faster.

5.3. Applying for CVRP. In this section, we apply APPE for solving CVRP. We use instances from [30, 31] to test APPE's performance. We compare APPE with DE, SSA, PSO, and PPE. For more effectively, we run 10 times for getting more reliable data. The BKV item means the best-known value of

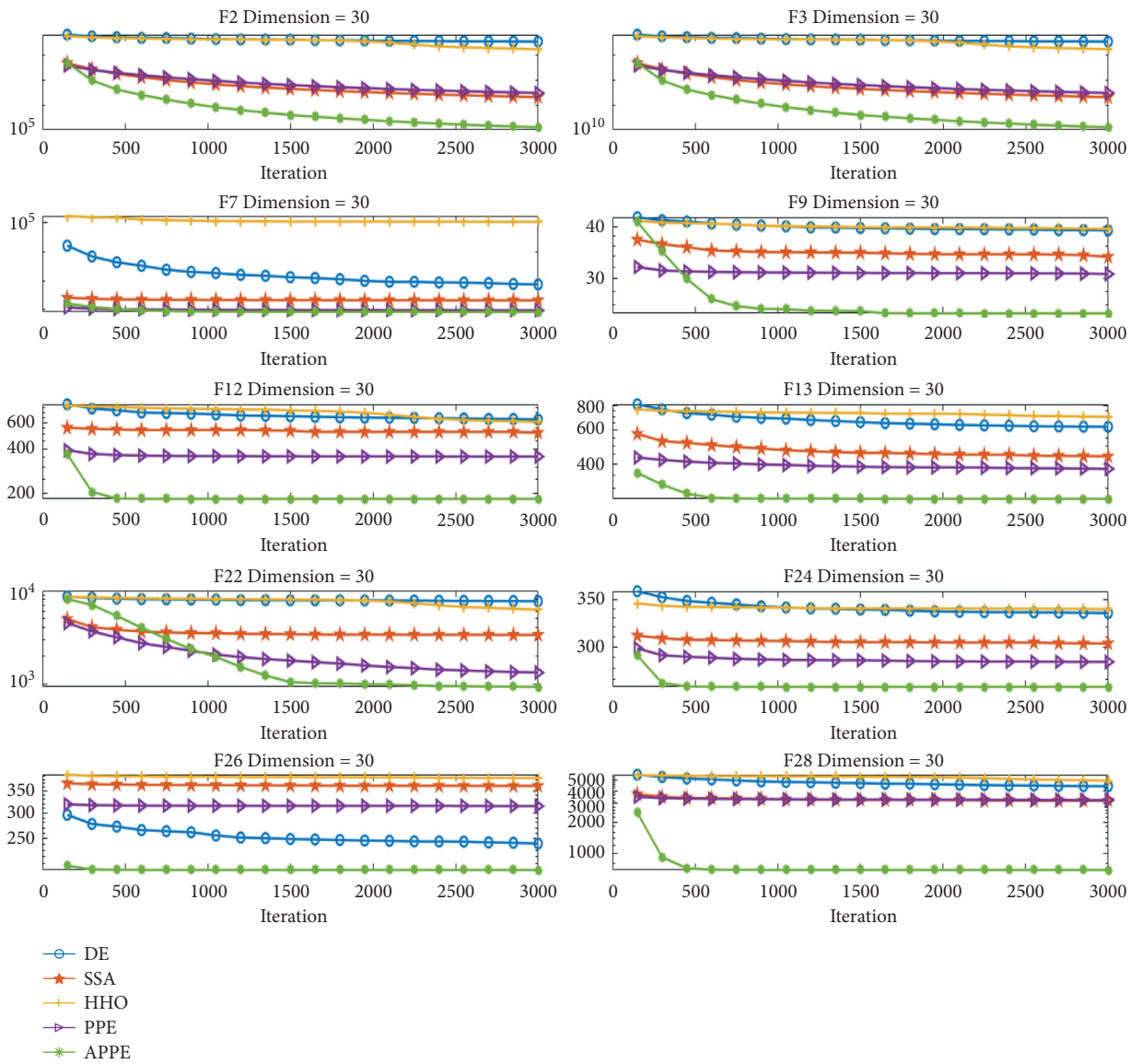


Figure 5: APPE's convergence curve with the dimension of 30.

instance. The particles are 50. The iteration is 1000. The setting of DE, PPE, and APPE are the same as aforementioned. For SSA, the number of the producers accounts for 20%, $SD = 20$, and $ST = 0.8$. For PSO, $\omega = 0.8$, and $c_1 = c_2 = 2$. We also use W/D/L to record each algorithm's win/draw/loss number in instances from Tables 14–19, and the compared fitness value is the sum of distances here. The CVRP experiment is shown in Tables 14 and 15.

In Table 14, obviously, the performance of APPE searching for the global optimum is better than that of the comparison algorithms. Most of the instances obtain solutions close to BKV. Similarly, in Table 15, APPE has a better convergence accuracy for CVRP than the contrast algorithms. Figure 4 is the best route result of A-n32-k5 solved by APPE. It has 5 routes, and its fitness is 787.0819 that is close to BKV. It is proved that APPE can effectively solve CVRP.

We also compare APPE with some existing results. Table 16 is the comparison results of APPE with the results of Korayem et al. [32]. The setting is consistent with [32] that the population is 20, the maximum generations are 500, and each instance runs 10 times. In Table 16, the convergence precision of APPE is better than KmeansFnO, KmeansFnP, and KmeansFnR that are all cluster-first route-second methods, and they combine k-means with gray wolf optimizer [32]. Table 17 shows the comparison results of APPE with the results of Yan et al. [33]. The setting is consistent with [33], and each instance runs 20 times. In Table 17, the convergence precision of APPE is better than the constraint optimization harmony search (CO-HS) of [33], PSO, and GA, and three comparison algorithm's data comes from [33]. Table 18 shows the comparison results of APPE with the results of Zhao et al. [34]. The setting is consistent with [34] that the population is 200, the maximum generations are 500, and each instance is run 10 times. It can be seen from Table 18 that APPE is superior to quantum DE (QDE), quantum evolution algorithm (QEA), and DE in both best and mean performance,

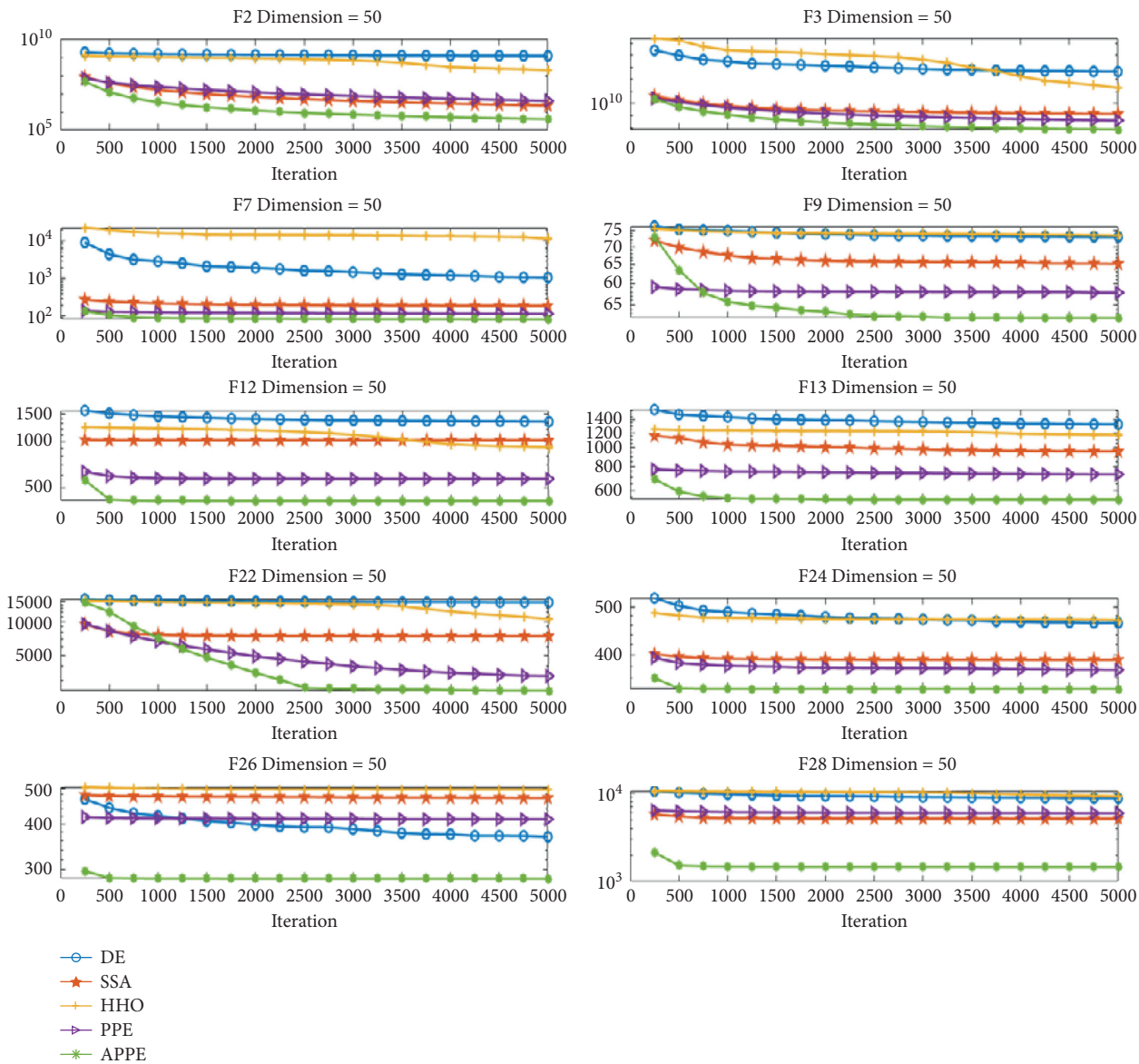


Figure 6: APPE's convergence curve with the dimension of 50.

with data from [34]. Table 19 shows the results of APPE and the work of Khairy et al. [22]. The setting is consistent with [22] that the population is 40, the maximum generations are 1000, and each instance is run 10 times. In Table 19, the convergence accuracy of APPE is significantly better than GA, ant colony optimization (ACO), and group teaching optimization (GTO), with data from [22].

6. Conclusions

We propose APPE that deletes competition and conditional acceptance and corresponding evolutionary trend update for shortening the algorithm's running time. It also adds a jump mechanism, history-based searching, and population closing moving for making PPE more likely to jump out of the local optimum and improving PPE's convergence accuracy. Then, we test APPE by CEC2013, which compares with DE, SSA, HHO, and PPE. Experiment results show that APPE has higher convergence accuracy and shorter running time. Finally, APPE is applied to solve CVRP. From the test results of instances, APPE is more powerful to solve CVRP than DE, SSA, PSO, and PPE. We also compare our algorithm with some existing work. The results show that APPE is able to solve CVRP.

In the future, APPE can be improved by hybrid other algorithm, such as Flower Pollination Algorithm,Harmony Search [35] and adding cubic chaotic mapping [36], a version of multi-objective APPE can be proposed by referring to inverse model [37]. APPE can also be applied to other fields, such as power system problems [38, 39], wireless sensor network problems [39, 40], dispatching system of public transit vehicles [41, 42], traffic forecasting [43], sensor ontology matching [44, 45], feature selection [46], surrogate approach [47, 48], and deep learning [49, 50].

Appendix

In the CEC2013 experiment, for the convergence curves of the three dimensions are approximate, we only put the convergence curves of $Di\ m = 10$ in the text and put $Di\ m = 30$ and $Di\ m = 50$ in the appendix. Figures 5 and 6 are the convergence curves of $Di\ m = 30$ and $Di\ m = 50$. It is similar to Figure 3, however, the convergence curve of APPE in Figures 5 and 6 has a relatively good convergence precision in the early stage of iteration. For Figures 5 and 6, in the middle of the iteration, APPE still continues to search at this time. In the latter part of the iteration, it enters the stable convergence stage like many algorithms. APPE's convergence curve is also basically at the bottom, i.e., the convergence precision is the best. To sum up, APPE has better convergence performance.

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61872085) and the Natural Science Foundation of Fujian Province (2018J01638).

References

[1] H. Ceylan, H. Ceylan, S. Haldenbilen, and O. Baskan, "Transport energy modeling with meta-heuristic harmony search algorithm, an application to Turkey," *Energy Policy*, vol. 36, no. 7, pp. 2527–2535, 2008.

[2] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," *International Journal of Advances in Soft Computing and Its Applications*, vol. 5, no. 1, pp. 1–35, 2013.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95 - International Conference on Neural Networks*, pp. 1942–1948, Perth, Western Australia, December 1995.

[4] J.-F. Chang, S.-C. Chu, J. F. Roddick, and J.-S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 809–818, 2005.

[5] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium Optimizer: A Novel Optimization Algorithm," *Knowledge-Based Systems*, vol. 191, 2020.

[6] J.-S. Pan, J. Zhuang, L. Liao, and S.-C. Chu, "Advanced equilibrium optimizer for electric vehicle routing problem with time windows," *Journal of Network Intelligence*, vol. 6, no. 2, pp. 216–237, 2021.

[7] X.-S. Yang, "Flower Pollination Algorithm for Global Optimization," in *Proceedings of the International Conference on Unconventional Computing and Natural Computation*, pp. 240–249, Orléans, France, September 2012.

[8] X.-S. Yang, *Nature-inspired Optimization Algorithms*, Elsevier, Chennai, India, 2014.

[9] J.-S. Pan, J. Zhuang, H. Luo, and S.-C. Chu, "Multi-group flower pollination algorithm based on novel communication strategies," *Journal of Internet Technology*, vol. 22, no. 2, pp. 257–269, 2021.

[10] J.-S. Pan, P. Hu, and S.-C. Chu, "Binary fish migration optimization for solving unit commitment," *Energy*, vol. 226, 2021.

[11] Z. Meng and J.-S. Pan, "QUasi-Affine TRansformation Evolution with External ARchive (QUATRE-EAR): an enhanced structure for differential evolution," *Knowledge-Based Systems*, vol. 155, pp. 35–53, 2018.

[12] T.-W. Sung, B. Zhao, and X. Zhang, "Quasi-affine transformation evolutionary with double excellent guidance," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 5591543, 2021.

[13] P.-C. Song, S.-C. Chu, J.-S. Pan, and H. Yang, "Simplified Phasmatodea Population Evolution Algorithm for Optimization," *Complex & Intelligent Systems*, 2021.

[14] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[15] J. K. Lenstra and A. H. G. R. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[16] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter, "On the capacitated vehicle routing problem," *Mathematical Programming*, vol. 94, no. 2-3, pp. 343–359, 2003.

[17] B. Wu, W. Wang, Y. Zhao, X. Xu, and F. Yang, "A Novel Real Number Encoding Method of Particle Swarm Optimization for Vehicle Routing problem," in *Proceedings of the 2006 6th World Congress on Intelligent Control and Automation*, pp. 3271–3275, Dalian, China, June 2006.

[18] A. Chen, G. Yang, and Z. Wu, "Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem," *Journal of Zhejiang University - Science*, vol. 7, no. 4, pp. 607–614, 2006.

[19] T. J. Ai and V. Kachitvichyanukul, "Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem," *Computers & Industrial Engineering*, vol. 56, no. 1, pp. 380–387, 2009.

[20] S. Z. Zhang and C. K. M. Lee, "An Improved Artificial Bee colony Algorithm for the Capacitated Vehicle Routing problem," in *Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2124–2128, Hong Kong, China, October 2015.

[21] A. M. Altabeeb, A. M. Mohsen, and A. Ghallab, "An improved hybrid firefly algorithm for capacitated vehicle routing problem," *Applied Soft Computing*, vol. 84, 2019.

[22] O. M. Khairy, O. M. Shehata, and E. I. Morgan, "Enhanced Group Teaching Optimization Algorithm for Solving the Capacitated Vehicle Routing Problem," in *Proceedings of the 2020 8th International Conference on Control, Mechatronics and Automation (ICCMA)*, pp. 222–226, Moscow, Russia, November 2020.

[23] Z. Fu, P. Hu, W. Li, J.-S. Pan, and S. Chu, "Parallel equilibrium optimizer algorithm and its application in capacitated vehicle routing problem," *Intelligent Automation and Soft Computing*, vol. 27, no. 1, pp. 233–247, 2021.

[24] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization," Technical Report, Nanyang Technol. Univ, Singapore, 2013.

[25] E. Atashpaz-Gargari and C. Lucas, *Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by*

Imperialistic Competition, pp. 4661–4667, IEEE congress on evolutionary computation, Singapore, 2007.

[26] R. A. Rutenbar, "Simulated annealing algorithms: an overview," *IEEE Circuits and Devices Magazine*, vol. 5, no. 1, pp. 19–26, 1989.

[27] L. Teng and H. Li, "A new frog leaping algorithm based on simulated annealing and immunization algorithm for low-power mapping in network-on-chip," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 3, pp. 716–722, 2018.

[28] M. Xiang and Q. Zhang, "Discrete cuckoo algorithm for capacitated vehicle routing problem," *Journal of Northeast Petroleum University*, vol. 45, no. 1, 2021.

[29] I Ilhan, "An Improved Simulated Annealing Algorithm with Crossover Operator for Capacitated Vehicle Routing Problem," *Swarm and Evolutionary Computation*, vol. 64, 2021.

[30] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi, *Computational Results with a branch and Cut Code for the Capacitated Vehicle Routing Problem*, Institut National Polytechnic, Dhaka, Bangladesh, 1995.

[31] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian, "New benchmark instances for the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 257, no. 3, pp. 845–858, 2017.

[32] L. Korayem, M. Khorsid, and S. S. Kassem, "Using Grey Wolf Algorithm to Solve the Capacitated Vehicle Routing Problem," in *Proceedings of the 3rd International Conference on Manufacturing, Optimization, Industrial and Material Engineering (MOIME 2015)*, Bali, Indonesia, May 2015.

[33] T. Yan, L. Wang, J. Zhou, and J. Wang, "An improved harmony search algorithm for CVRP," *Computer Technology and Development*, vol. 26, no. 9, pp. 187–191, 2016.

[34] Y. Zhao, H. Jiang, and J. Zhang, "A quantum differential evolution algorithm for the vehicle routing problem," *Journal of Zhejiang University of Technology*, vol. 48, no. 1, pp. 68–72+111, 2020.

[35] O. Abdel-Raouf, I. El-Henawy, and M. Abdel-Baset, "A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles," *International Journal of Modern Education and Computer Science*, vol. 6, no. 3, pp. 38–44, 2014.

[36] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *Journal of computational design and engineering*, vol. 5, no. 4, pp. 458–472, 2018.

[37] X. Xue, C. Jiang, H. Wang, P.-W. Tsai, G. Mao, and H. Zhu, "An Improved Multi-Objective Evolutionary Optimization Algorithm with Inverse Model for Matching Sensor Ontologies," *Soft Computing*, vol. 25, 2021.

[38] H. M. Dubey, M. Pandit, and B. K. Panigrahi, "A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems," *Cognitive Computation*, vol. 7, no. 5, pp. 594–608, 2015.

[39] X. Cheng, Y. Jiang, D. Li, Z. Zhu, and N. Wu, "Optimal operation with parallel compact bee colony algorithm for cascade hydropower plants," *Journal of Network Intelligence*, vol. 6, no. 3, pp. 440–452, 2021.

[40] J. Wu, M. Xu, F.-F. Liu, M. Huang, L.-H. Ma, and Z.-M. Lu, "Solar wireless sensor network routing algorithm based on multi-objective particle swarm optimization," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 12, no. 1, pp. 1–11, 2021.

[41] X. Dong, Y. Zhang, and S. Yu, "An uneven clustering routing protocol based on improved K-means algorithm for wireless sensor network in coal-mine," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 1, pp. 53–62, 2019.

[42] M. Zhu, S.-C. Chu, Q. Yang, W. Li, and J.-S. Pan, "Compact sine cosine algorithm with multigroup and multistrategy for dispatching system of public transit vehicles," *Journal of Advanced Transportation*, vol. 2021, Article ID 5526127, 2021.

[43] R. L. Abduljabbar, H. Dia, P.-W. Tsai, and S. Liyanage, "Short-term traffic forecasting: an LSTM network for spatial-temporal speed prediction," *Future Transportation*, vol. 1, no. 1, pp. 21–37, 2021.

[44] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6625184, 2021.

[45] S.-C. Chu, X. Xue, J.-S. Pan, and X. Wu, "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.

[46] Y. Zhang, Y. Liu, and C.-H. Chen, "Review on deep learning in feature selection," in *Proceedings of the 10th International Conference on Computer Engineering and Networks*, pp. 439–447, Xi'an, China, January 2020.

[47] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.

[48] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, "Surrogate-assisted hierarchical particle swarm optimization," *Information Sciences*, vol. 454-455, pp. 59–72, 2018.

[49] Y. Zhang, Y. Liu, and C.-H. Chen, "Survey on Blockchain and Deep Learning," in *Proceedings of the IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1989–1994, Guangzhou, China, December 2020.

[50] L. Wu, C.-H. Chen, and Q. Zhang, "A mobile positioning method based on deep learning techniques," *Electronics*, vol. 8, no. 1, 2019.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 7891475, 10 pages
https://doi.org/10.1155/2021/7891475

Research Article

Modeling and Simulation of Wake Safety Interval for Paired Approach Based on CFD

Xin He,[1] Yilong Ma,[2] Hong Yang,[3] and Yaqing Chen[3]

[1]*Air Traffic Management College, Civil Aviation Flight University of China, Guanghan 618307, China*
[2]*Hope College, Southwest Jiaotong University, Chengdu 610400, China*
[3]*Civil Aviation Flight Technology and Flight Safety Research Base, Civil Aviation Flight University of China, Guanghan 618307, China*

Correspondence should be addressed to Yaqing Chen; chenyaqingmail@sina.com

Received 16 July 2021; Accepted 29 November 2021; Published 30 December 2021

Academic Editor: Chi-Hua Chen

In order to relieve the stress caused by the surge of flight flow, Closely Spaced Parallel Runways (CSPRs) have been built in many hub airports, and a paired approach mode has been applied to CSPRs in some countries. This paper proposes a method for optimizing the wake separation between aircrafts which utilizes a paired approach, aiming at reducing longitudinal separation by using computational fluid dynamics technology. Firstly, the model of the wake vortex field of the paired lead aircraft is constructed. Secondly, the numerical simulation preparation for the characteristics of the wake vortex field is completed through the computational pretreatment of the model. Thirdly, a calculation model of wake safety interval based on paired approach operation is established. Finally, the proposed method shows its superiority comparing with other methods. This method realized visual analysis of wake vortex through optimization modeling based on computational fluid dynamics, contributing to increasing the capacity of the runway and improving the operation efficiency of an aerodrome.

1. Introduction

In recent years, the airline industry develops rapidly and plays an important role in the world economy. The increasing trend in civil aviation transportation has remained high for years, and airport capacity has begun to show signs of decline. Most aviation hub cities have plans to rebuild and expand airports, so some large hub airports began to build closely spaced parallel runways (CSPRs).

Regarding the operation of CSPRs, due to safety considerations, the wake interval standard for single-runway operation is still adopted in China for the time being, which has not stimulated the expansion advantage of CSPRs. However, the paired approach (PA) mode was proposed by American scholars. The PA concept is the one that leverages the real-time navigation and communication capabilities of the ADS-B equipage initiative to increase airport capacity by performing simultaneous dependent approaches to CSPRs [1, 2]. This operation mode greatly increases the capacity of CSPRs when successive aircrafts are approaching under instrument meteorological conditions, which greatly reduces the wake interval and avoids the wake influence caused by the lead aircraft, providing the operation of CSPRs with a new direction [3].

In 2000, The concept of "safety zone" was introduced by Landry and Pritchett through the research of the paired approach [4]. The safety zone, also called the protection zone, of the paired approach mode is shown in Figure 1. It is an area in which the paired aircrafts can maneuver operate, and separation assurance and wake avoidance are provided. The front boundary of safety zone called collision safety limit (CSL). The rear boundary of the safety zone called wake safety limit (WSL). The American Mitte Company proposed a 3° offset paired approach, which can better avoid the risk of wake encounters using computational fluid dynamics [5]. This research result has played a great role in the research of the subsequent paired approach mode, which has provided theoretical guidance for a simultaneous instrument

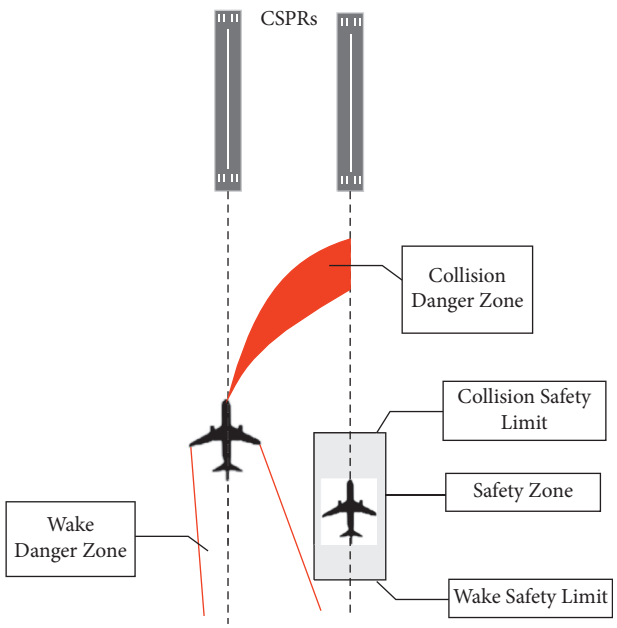


Figure 1: Schematic diagram of paired approach mode.

approach mode implemented at the airports such as Boston and San Francisco [6–8]. In 2001, the Advanced Aviation System Development Center of the company and the Industrial and System Engineering Campus of Georgia Tech University tested the initial procedures for the paired approach on a flight simulator, laying a foundation for the further definition of cockpit missions and the development of a cockpit separation management system [9].

NASA proposed to use the Monte Carlo method for the simulation calculation of the wake encounter risk caused by lead aircraft and proved the applicability of the method. The safety zone of the paired approach and the risk of collision during a wrong approach of the lead aircraft can also be simulated by using the Monte Carlo method [10–12]. In 2014, the Langley Research Center used the method of constructing a wake kinematics model to calculate the wake encounter risk for aircrafts that implemented the paired approach mode, which provided a reference for the quantitative analysis.

With the development of new technologies in civil aviation communications, navigation, surveillance equipment, and air traffic control, the operation of the paired approach mode has become more safe and efficient with the support of advanced equipment such as ADS-B surveillance equipment and the next-generation warning system ALAS. This has opened the door for the paired approach procedure based on interval management [13–16]. The Flight Research and Analysis Group of FAA began a safety study on the collision risk in 2017, associated with the lead aircraft's front boundary of the safety zone [2]. From 2018 to 2019, funded by NASA, MOSAIC ATM, United Airlines, and Honeywell, the research group jointly planned and conducted a paired approach mode flight test at San Francisco International Airport. The data obtained from the test will provide for the further implementation of the paired approach mode [17, 18].

This paper considered the safety of the paired approach mode, proposing a new method to study the wake safety interval of the paired aircrafts. We build the optimization model of the wake vortex field of the paired lead aircraft so that we can achieve visualized analysis of the wake influence. Compared with previous studies, the innovations of this paper are as follows:

Compared with experimental methods—wind tunnel test and water tunnel test, the CFD numerical simulation method shortens the research time, because it can rely on the rapid computing ability of the computer to build a vortex field model and carry out the numerical simulation, rather than constructing, testing, and maintaining the experimental equipment.

The CFD numerical simulation method not only improves the accuracy to a certain extent but also adapts to the numerical simulation of complex vortex fields.

Using the CFD method to study the safety zone of the paired approach wake is a breakthrough attempt, which provides new ideas and new solutions for using the fluid mechanics method to study the actual operation of civil aviation.

The authors' main contributions are summarized as follows: X.H. and Y.C. conceptualized and supervised the study; X.H. developed methodology and performed formal analysis; H.Y. collected data and wrote the original draft; Y.M. investigated and developed the model by software; Y.C. and X.H. validated the study; Y.C. collected resources and funds; all authors have read and agreed to the published version of the article.

2. Research Method of Wake

To determine the wake influence range of the paired approach aircraft, the motion characteristics of the wake vortex field of the paired aircraft must be analyzed and studied. At present, the research on the motion characteristics of aircraft wake vortex field mainly includes the following methods: wake feature detection test method, wake feature capture modeling method, and wake numerical simulation method. This section will discuss in detail the advantages and disadvantages of these methods.

2.1. The Test Method of Wake Feature Detection. The wake vortex has unique electromagnetic scattering characteristics, so it is easy to be effectively detected by sonar, radar, lidar, and other sensors [19]. Carrying out the wake feature detection test is of great value to the research and application of military and civil aviation.

Wind tunnel test is a conventional method to study the characteristics of the wake field of an aircraft. This method can measure the velocity distribution of the wake field in a certain space behind the aircraft model under different configurations. However, this method is subject to the influence of the test section scale and is mainly aimed at the detection of the characteristics of the wake field near the aircraft. The process of the wake generation and rolling up is shown in Figure 2 [20]. Measurement methods usually

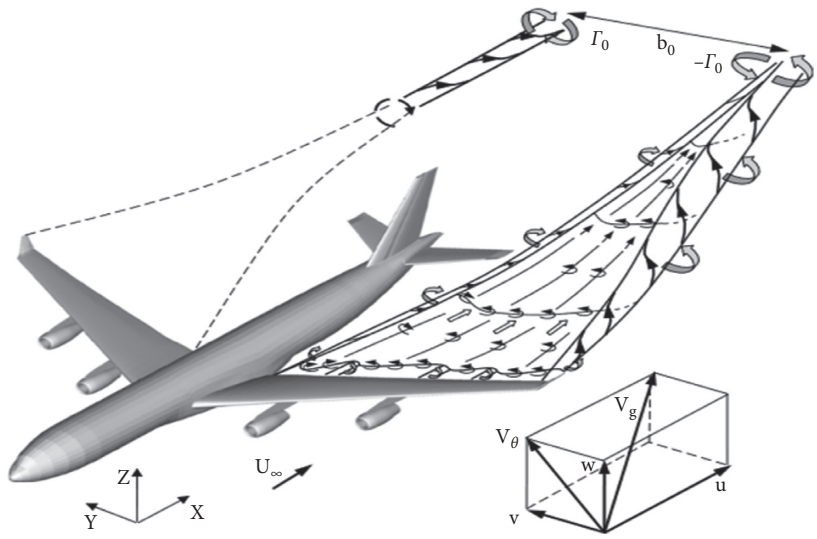


Figure 2: Wake generation and winding process.

include five-hole probe sensor measurement method [21], particle image velocimetry measurement method [22], or hot wire wind velocity method for measuring the wake field velocity over time. Since the 1970s, a large number of documents have recorded the velocity and vorticity fields of various configurations of typical transport aircraft measured by the wind tunnel test [23–31]. This method has the characteristics of good vortex field quality and high measurement accuracy, but it also has the following shortcomings: the wind tunnel structure is relatively complicated, the manufacturing cost is high, the experiment area is large, and it relies on high-precision sensors.

The water tunnel is a common test which uses water as the medium for dynamic experiments. The water tunnel test is more suitable for the experimental observation and research analysis of certain aerodynamic problems than the wind tunnel test, such as the generation and shedding of aircraft wingtip vortices [32]. The advantage of the water tunnel test is that the multiscale vortex structure near the wall, the wake vortex field shedding structure, and the spatial vortex separation structure can be clearly understood. The observed vortex field structure will not be interfered with by the tracer particle material properties, and the effect will be impaired [33, 34]. In addition, the water tunnel experiment also has the following shortcomings: the water tunnel structure is relatively complex, the manufacturing cost is high, the research period is long, and it is easy to cause experimental system errors due to the interference of the subsystems in the vortex field and the influence of the fluid medium or model. Especially most experiments use scaled models for experiments, and the simulated vortex environment cannot be absolutely consistent with the actual working conditions.

2.2. The Method of Wake Feature Capture Modeling. In order to further simplify the analysis process and enhance the prediction effect, scholars have improved the modeling method of the wake vortex field based on the observations obtained from flight tests. Since these wake velocity field modeling methods are based on the theoretical analysis methods of fluid mechanics, they are very effective for some uncomplicated vortex problems. However, for some more complex and nonlinear governing equations, they are beyond reach. Thus, it is suitable for qualitative analysis of simple vortex problems or complex fluid mechanics problems.

2.2.1. Lift Line Theory and Wake Vortex Hypothesis. From the classic Prandtl lift line theory, it can be known that if the wingspan of an airplane wing is constant, when it generates lift power, a pair of vortices in opposite directions will be derived behind it. The relevant variables of the above-mentioned reverse vortex pair can be calculated with the relevant performance data of the aircraft (maximum take-off weight, flight speed, wingspan, etc.). As a result, the calculation of wake development and evolution has been simplified. And this theory and hypothesis have been widely used in the study of wake analysis.

$$b_0 = \frac{\pi}{4} b,$$
$$r_c = 005 b_0, \tag{1}$$
$$\Gamma_\infty = \frac{W_{T_0} g}{\rho \cup b_0},$$

where b_0 is the distance between the left and right vortices; b is the wingspan length; r_c is the vortex core radius; Γ_∞ is the vortex ring volume; W_{T_0} is the maximum take-off weight; ρ is the air density; $\cup$ is the acceleration of gravity; $\cup$ is the velocity of the incoming vortex.

2.2.2. Convection-Diffusion Equation. The convection-diffusion equation is a nonlinear equation used to characterize convection and diffusion. And it is also a kinematic

equation, which is generally used in areas where the wake velocity is relatively stable, such as near and middle wakes. The mathematical expression when using the convection-diffusion equation for time-domain simulation is as follows [35]:

$$\begin{cases} \dfrac{\partial s}{\partial t} + u_i \dfrac{\partial s}{\partial x_i} - D \dfrac{\partial^2 S}{\partial x_2^2} = 0, \\ \\ S[y\hat{y}, 0] = y_0 - y, \end{cases} \tag{2}$$

where u_i is the rotational velocity field in the wake; D is the diffusion coefficient of air; S is the conservative passive quantity of the solution of the equation; $[y\hat{y}, 0] = y_0 - y$ is the initial condition.

2.3. Numerical Simulation of Aircraft Wake Based on CFD. In order to make up for the difficulties such as high cost, long research time, and low accuracy of the test methods, the CFD method began to be applied to the analysis and prediction of the wake vortex field. In the 1960s, Takami used the vortex plate method for the numerical simulation of the wake, which can obtain the induced velocities at various locations in the vortex field [36]. Scholars visualized the time-based simulation and formed a wake model that can vary with a scale, which officially started the real three-dimensional wake simulation [37]. When the Reynolds average N-S equation appeared and began to be applied, CFD numerical methods began to be used to simulate the three-dimensional vortex field from the generation, development, and dissipation of wake [38]. Recent studies have shown a variety of numerical simulation methods, including large eddy simulation, which show good vortex characteristics for wake simulations, that is in good compliance with some phenomena observed in experiments [39].

In addition, the CFD numerical simulation method can also achieve faster simulation by combining classic wake models without the excessive pursuit of calculation accuracy. The initial wake vortex field distribution data obtained by the wake model can further develop the numerical evolution, which can predict the safety zone of the wake [39]. On the basis of retaining a certain calculation accuracy, the calculation efficiency is greatly improved, which is of great significance to the calculation and pre diction of wake interval. The CFD method has unique advantages such as short calculation time, low cost, easy data extraction, and the continuous optimization and improvement of CFD technology, making it the main direction of fluid mechanics research.

3. CFD Modeling of Aircraft Wake Vortex Field

In order to analyze the motion characteristics of the wake vortex field of the paired aircrafts, the CFD method is used to construct the wake vortex field model of the paired aircrafts. The numerical simulation process of the wake vortex field is completed through preprocessing of the model calculation and setting of boundary conditions. Finally, by postprocessing the numerical calculation results, the wake vortex field of the paired aircrafts can be quantitatively analyzed.

3.1. Model Construction of Aircraft Wake Vortex Field. The construction of the aircraft wake vortex field model is an important part of the numerical simulation process. The density, quantity, and quality of the grid cells in the built model will directly affect the accuracy of the numerical simulation. The construction of the aircraft wake vortex field model mainly includes the following steps: establishment of the aircraft wing geometric model, construction and design of the aircraft wake vortex field calculation domain, and model calculation preprocessing (mesh division, dense calculation domain, setting boundary conditions).

3.1.1. Geometric Modeling of the Lead Aircraft Wing. In view of the extremely complex configuration of the aircraft wing, the entire wing is composed of several parts connected together. In addition to the main wing, there are multiple structures such as flaps, ailerons, spoilers, winglets, and engine pods. If the real wing structure is to be simulated numerically, it is not only difficult to model but also difficult to realize numerical simulation with the existing computer technology and CFD technology. Therefore, this paper simplifies the actual physical configuration of the wing. Based on the standard model of Boeing's transport airliner, the modeling tool SolidWorks is used to intercept and construct the wing geometric model. According to the actual geometric size of the paired aircrafts, the model is scaled down to obtain a suitable geometrical model of the wing. Taking the B747-400 wing as an example, the wingspan of the B747-400 is 211 ft, and the reduced geometric model of the B747-400 wing is shown in Figure 3. During an aircraft approach, it usually flies at a certain angle of attack, and a certain angle of attack can be set when the model is built.

3.1.2. Calculation Domain Setting of Aircraft Wake Vortex Field. The computational domain of the vortex field refers to the mathematical operation (usually integral operation) during the numerical simulation calculation process. There are two ways to generate the computational domain—direct modeling and geometric extraction. In this paper, we need to study the wake vortex field of aircraft. The scale of the wake vortex is relatively large, so it is obviously impossible to generate the computational domain through geometric extraction. It is more appropriate to adopt direct modeling. In the actual calculation, in order to reduce the calculation load and the calculation pressure, the geometry of the calculation domain is usually simplified. Using the symmetry of the geometric model and the periodicity of fluid flow are common geometric simplification methods. For the numerical simulation of the external vortex field of the aircraft, the experimenter usually constructs the computational domain as a cylinder. In order to reduce the amount of calculation and take into account the motion characteristics of the wake, this paper simplified the calculation domain of

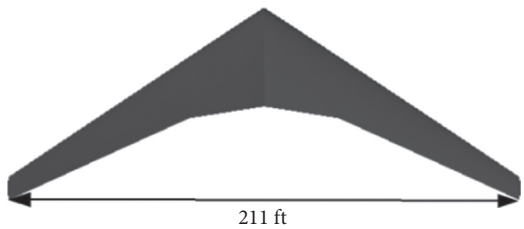


Figure 3: Geometric model of B747-400 wing.

the outer vortex field of the aircraft to the truncated cone shown in Figure 4. The calculation domain of the outer vortex field is divided into INLET, OUTLET, and WALL.

3.1.3. Calculation Model Preprocessing. The preprocessing process of the model generally includes meshing of the vortex field model, compacting the computational domain, and setting the boundary conditions. Compared to the construction and design of the computational domain, meshing is more important. The proper processing of the computational grid and the quality of grid generation are the primary conditions for numerical simulation calculations. The selection of grid quantity is generally based on empirical value or reference to the recommendations of relevant literature. In this paper, we use ANSYS ICEM CFD for preprocessing. It can provide advanced grid generation, geometry acquisition, and grid cell optimization functions for the analysis of complex models. It can also output grids for fluid mechanics solvers. The modeling tool SolidWorks will be used to intercept and build the wing geometric model. Then, it is imported into ANSYS ICEM CFD, the geometric model is checked, geometric repair is performed, and the above-mentioned calculation domain is generated. Then, the appropriate grid algorithm is selected to generate the computational grid of the geometric wing model and the computational domain, that is, the computational grid of the wake vortex field model. There are three grid algorithms provided by ANSYS ICEM CFD such as Octree algorithm, Delaunay algorithm, and Advance Front algorithm. The octree algorithm is commonly used for the formation of unstructured grid tetrahedron. This algorithm is more intelligent and automatic, so it was chosen for grid generation in this paper. In order to save computing resources and time, an unstructured grid is selected in this paper. The process of using ANSYS ICEM CFD for meshing is shown in Figure 5 in detail. First, whole grid cells are set, including grid parameters, plane grid, volume grid, prism layer, and period. Then, grid encryption is carried out for the computing domain by setting partial grid, plane grid, and line grid, and creating grid density region. Finally, meshing these grids, including three parts: meshing plane grid, meshing volume grid, and meshing prism layer grid. After the above steps, the wake vortex field of the paired aircrafts was meshed. The mesh model of the wing geometry model and the computational domain mesh model after the division are shown in Figure 6 and Figure 7, respectively. The X-axis indicates the direction of the incoming vortex, the Y-axis indicates the span direction, and the Z-axis indicates the height. The total

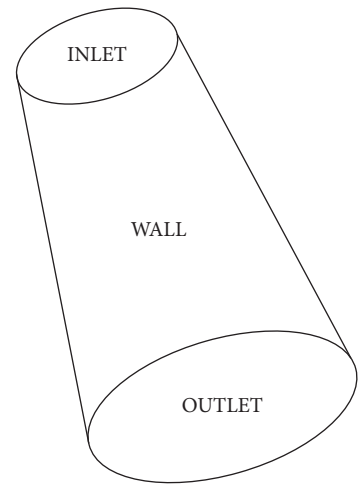


Figure 4: Computational domain of aircraft wake vortex field.

grid quantity of the wake vortex field is about 14,000,000. Taking into account the principle of the wake generation during aircraft flight, the leading edge of the wing needs to be arranged with a dense grid so as not to affect the calculation accuracy. In order to capture the wingtip vortices, the computational domain grid of the area where the wake is formed downstream of the aircraft should also be for refinement processing. The grid division details of the leading edge of the wing are shown in Figure 8, and the details of computational domain grid refinement are shown in Figure 9.

For the finished grid, it cannot be directly used for calculation but must go through grid output setting and grid quality check. The grid quality is related to the calculation precision and also determines whether the calculation can converge. Check and Report Quality functions in ANSYS ICEM CFD can be used to check the grid. It mainly detects the geometric size, volume, and grid proportion of the mesh model, among which the minimum volume parameter needs to be paid more attention, and its value must be positive. When the grid model has a negative volume, it cannot be used for numerical simulation calculations. It is generally believed that the grid slope is better than 0.3. the Check and Report Quality functions are used in ANSYS ICEM CFD software to conduct a quality check on the grid model. The quality check result of the grid is shown in Figure 10. A small part of the grid has a slope of less than 0.3, and there is no negative volume grid. It is considered to meet the requirements of numerical simulation calculation.

4. Calculation of Wake Separation Based on CFD

There are two safe areas in the paired approach mode. They are the first safe area (the safety zone mentioned in Section 2) and the second safe area. This article focuses on the first safe area in the paired approach mode. The first safe area is the maneuvering flight area that is between the CSL and WSL of the paired aircrafts. The second safe area is an area where the wake of the lead aircraft has dissipated. Due to the

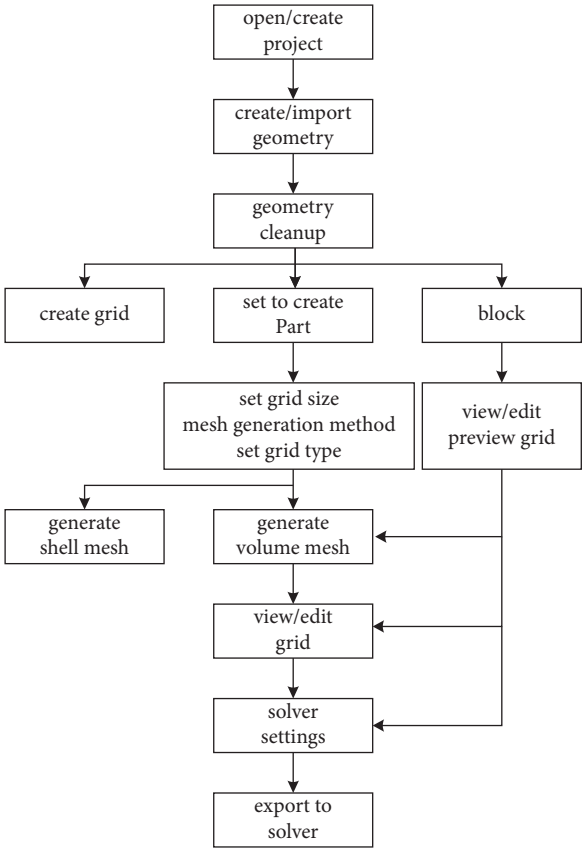


Figure 5: ANSYS ICEM CFD meshing process.

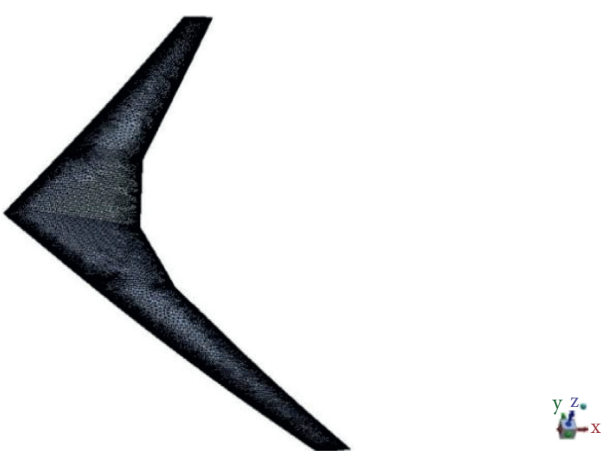


Figure 6: Mesh model of the wing geometry model.

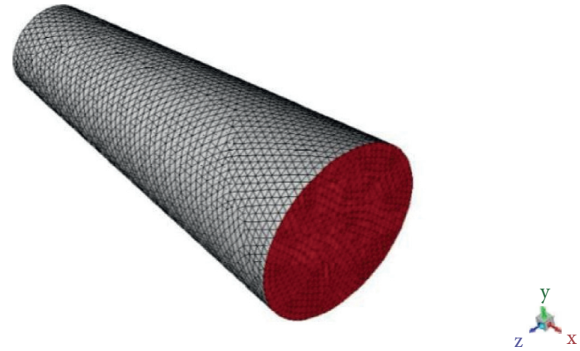


Figure 7: Computational domain grid model.

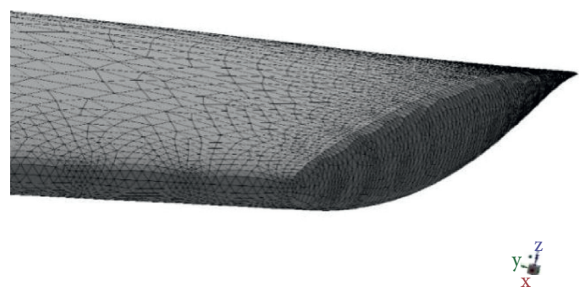


Figure 8: Details of the meshing lead edge of the wing.

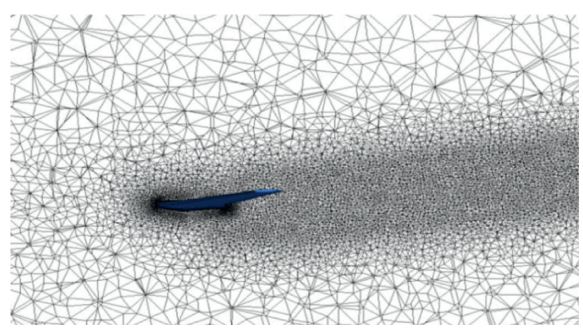


Figure 9: Detailed diagram of computational domain grid encryption.

development of communication, navigation, and surveillance equipment, the CSL caused by the wrong approach of the paired lead aircraft can basically be ignored. Therefore, the research on the first safe area (safety zone) of the paired approach mode can focus on the WSL which needs to calculate the maximum wake safety interval.

The CFD-based wake safety interval calculation method combines knowledge and technology in the fields of fluid mechanics, mathematics, and computer science. The purpose of this method is to make up for the difficulty of the test method, the high cost, and the long research period. It can also make up for the shortcomings of using the wake model method to simulate complex wake vortex fields with insufficient accuracy. The calculation of the wake safety separation tends to be simplified under the premise of ensuring a certain calculation accuracy.

4.1. Determining the Initial Lateral Interval of the Paired Aircraft. For the 3° offset paired approach mode, before the trailing aircraft reaches the wake protection point, there is always a vertical safety gap of 1,000 ft between the two aircrafts, so the trailing aircraft does not need to consider the impact of the wake of the lead aircraft. When the trailing aircraft reaches the wake protection point, since the paired aircrafts do not need to maintain a vertical separation of

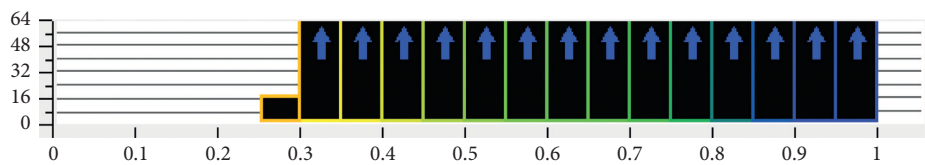


Figure 10: Grid quality inspection results.

1,000 ft at this time, the trailing aircraft must consider avoiding the influence of the wake of the lead aircraft. For the straight-in paired approach mode, since the distance between the runway centerlines of CSPRs is less than 2500 ft, a safe wake distance between the two aircraft should be determined at the beginning of the pairing to ensure flight safety. At this time, the initial lateral separation between the two aircraft is the distance between the centerline of the runway (C). Since the wake of the lead aircraft is generated near the wingtip, in order to make the calculation result more conservative, one half of the wingspan of the rear aircraft (B_2) should be deducted from the runway centerline spacing (D_0) as the starting side of the paired aircraft which is shown in Figure 11.

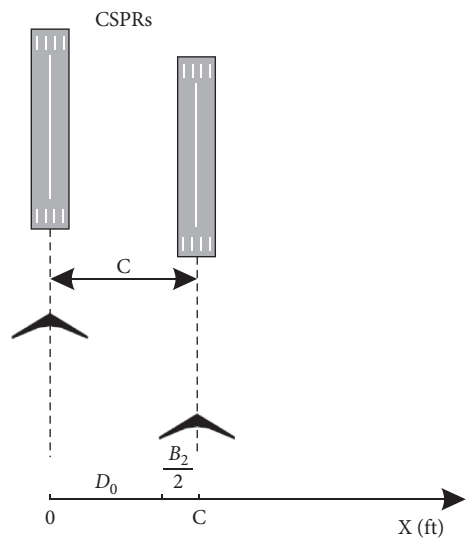


Figure 11: The initial lateral interval of the paired aircrafts.

4.2. Numerical Simulation of the Lead Aircraft's Wake Vortex Field. According to the methods and steps described above, the wake vortex field of the lead aircraft is numerically simulated. The result file is obtained after the numerical simulation calculation converges. The result file of the numerical simulation of the lead aircraft wake vortex field is postprocessed to obtain a visual lead aircraft wake vortex field (vorticity cloud map, vorticity isosurface map, etc.), as shown in Figure 12. Combining the basic process of the paired approach mode, the CFD-based paired approach wake safety interval calculation model is constructed.

Wing-tip vorticity, which is the wake of aircraft during flight, can be identified by vorticity. In TecPlot and CFD-Post, the variable vorticity needs to be customized. For example, in CFD-Post, new expressions and new variables can be created, and the two corresponding velocity differential variables can be loaded successively. The variable vorticity can be obtained by subtraction operation, and then the vorticity cloud map can be obtained. In TecPlot, the variable vorticity can also be obtained by creating a vorticity calculation formula in a similar way. The specific formula is as follows [24]:

$$\begin{aligned} \{wx\} &= ddy(\{w\}) - ddz(\{v\}), \\ \{wy\} &= ddz(\{u\}) - ddz(\{v\}), \\ \{wz\} &= ddx(\{v\}) - ddz(\{v\}). \end{aligned} \tag{3}$$

4.3. Determining the Wake Safety Interval. As mentioned in the previous section, the strength and direction of the vortex can be described by the vorticity, so this article uses the vorticity as the physical quantity to identify the range of the wake vortex field. The paired approach mode requires the trailing aircraft to avoid the wake of the lead aircraft before the wake of the lead aircraft arrives so as to keep the trailing aircraft always maneuvering in the first safety zone. Therefore, according to the CFD-based paired approach wake safety interval calculation model, the maximum wake safety interval between the paired aircrafts can be obtained. The wake safety interval (D) at this time is obtained from the vorticity distribution on the XY section ($Y = D_0$) of the lead aircraft in the downwind direction. Assuming that the vorticity is on the XY section of the downwind direction of the paired aircraft, at the time $Y = D_0$ and point $X = D_1$, the absolute value is greater than 0, and the wake safety interval (D) of the paired aircraft can be obtained as follows:

$$D = D_1. \tag{4}$$

4.4. The Limitation of Present Work. This model can provide some theoretical reference for the operation of paired approach mode. However, the following aspects need further discussion and research: the wing model of the paired lead aircraft established in this paper is the standard wing model of transport aircraft. In future research, a specific model can be developed for each type of aircraft, to further reduce the error between numerical simulation test and real value. When constructing the calculation model of safe wake interval based on CFD, it is assumed that the distance between two aircrafts is the maximum safe wake interval when the absolute value of vorticity at the wing of the trailing aircraft is not zero, which is just to simplify the

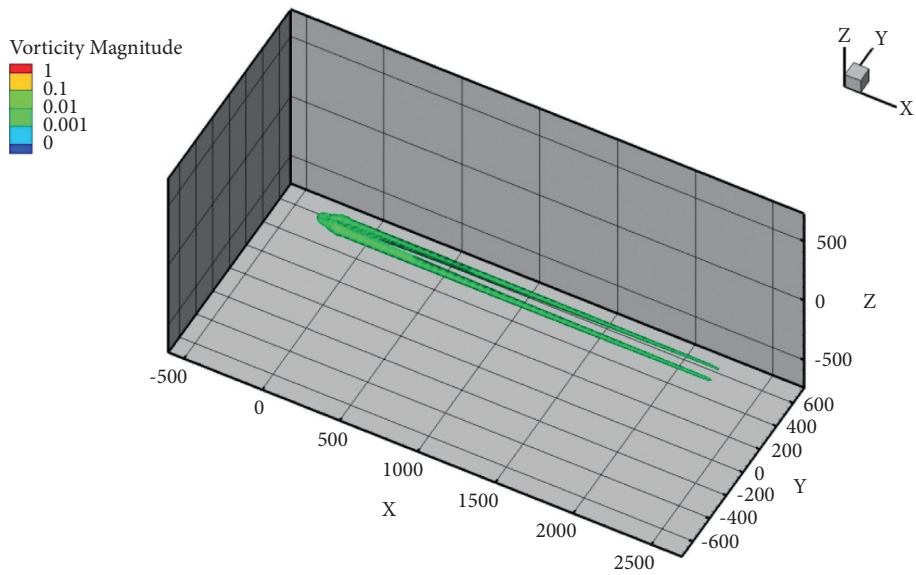


Figure 12: Isosurface map of the tail vortex of the paired aircraft with a vortex of 0.01 under static wind conditions.

calculation. In the future, the trailing aircraft can be assumed as a cuboid or ellipsoid to make it a more reasonable model.

5. Conclusion

In this paper, the computational fluid dynamics method is used to model the wake separation optimization when two aircrafts are operating in the paired approach mode. Through the theory introduction, modeling introduction in detail, and numerical simulation and analysis concluded that using the CFD method to study the safety zone of the paired approach wake is feasible. Comparing with experimental methods—wind tunnel test and water tunnel test, the CFD numerical simulation method shows great superiority. The research results of this paper can provide a certain theoretical reference for the study of the wake safe area of the paired approach mode and provide certain theoretical support for the specific implementation of the paired approach mode in domestic air transportation.

This method also has some shortcomings that need to be improved. This method only considered the first safe area, ignoring the collision risk caused by the lead aircraft when it makes a wrong approach and intrudes in the trailing aircraft's course. This would be the future research direction of the study. Also, a comparative study between experimental methods and CFD would be considered in the follow-up study.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Key Laboratory of Civil Aviation Flight Technology and Safety, CAAC Aviation Safety Capacity Building Fund Supported Project and CAAC Air Traffic Management Bureau Project.

References

[1] F. Holzäpfel, T. Gerz, F. Köpp et al., "Strategies for circulation evaluation of aircraft wake vortices measured by lidar," *Journal of Atmospheric and Oceanic Technology*, vol. 20, no. 8, pp. 1183–1195, 2003.

[2] M. L. Williams, L. C. Wood, and B. J. Nelson, "Safety study of closely spaced parallel operations utilizing paired approach," in *Proceedings of the2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, pp. 1–10, IEEE, San Diego, CA, USA, September 2019.

[3] M. C. Waller and C. H. Scanlon, "Flight deck centered parallel runway approaches in instrument meteorological conditions," in *Proceedings of the NASA workshop on flight deck centered parallel runway approaches in instrument meteorological conditions*, Washington, DC, USA, October 1996.

[4] S. J. Landry and A. R. Pritchett, "Incorporating procedural information in a paired approach task," *Proceedings of the Human Factors and Ergonomics Society - Annual Meeting*, SAGE Publications Sage CA, vol. 46, no. 1, pp. 106–110, 2002.

[5] R. R. Eftekari, J. B. Hammer, D. A. Havens, and A. D. Mundra, "Feasibility analyses for paired approach procedures for closely spaced parallel runways,"in *Proceedings of the IntegratedCommunications, Navigation, and Surveillance Conference Proceedings*, IEEE, Herndon, VA, USA, May 2011.

[6] J. Hammer, "Case study of paired approach procedure to closely spaced parallel runways," *Air Traffic Control Quarterly*, vol. 8, no. 3, pp. 223–252, 2000.

[7] R. Bone, O. Olmos, A. Mundra, and R. Jensen, *Paired Approach: A Closely Spaced Parallel Runway Approach Concept*, Federal Aviation Administration, Washington, DC, USA, 2000.

[8] R. Teo, J. S. Jang, and C. Tomlin, "Flight demonstration of provably safe closely spaced parallel approaches," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 6197, San Francisco, CA, USA, August 2005.

[9] R. Bone, A. Mundra, and B. O. Olmos, "Paired approach operational concept," in *Proceedings of the 20th DASC. 20th Digital Avionics Systems Conference(Cat. No. 01CH37219)*, vol. 1, pp. 5B3/1–5B3/14, IEEE, Daytona Beach, FL, USA, October 2001.

[10] A. Pritchett, S. Landry, and A. Pritchett, "Two studies of paired approaches," in *Proceedings of the Air Traffic Management Research and Design Workshop*, Sante Fe, NM, USA, November 2001.

[11] B. McKissick, "Wake encounter analysis for a closely spaced parallel runway paired approach simulation," in *Proceedings of the 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium*, p. 6943, Hilton Head, SC, USA, October 2009.

[12] N. Guerreiro, K. Neitzke, S. Johnson, H. Stough, B. McKissick, and H. Syed, "Characterizing a wake-free safe zone for the simplified aircraft-based paired approach concept," in *Proceedings of the AIAA Atmospheric and Space Environments Conference*, p. 7681, Toronto, Canada, August 2010.

[13] M. M. Madden, "Kinematic modeling of separation compression for paired approaches to closely-spaced parallel runways," in *Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3150, Atlanta, GA, USA, June 2014.

[14] S. C. Johnson, G. W. Lohr, B. T. McKissick, N. M. Guerreiro, and P. Volk, "Simplified aircraft-based paired approach: concept definition and initial analysis," National Aeronautics and Space Administration, Langley Research Center, Hampton, VA, USA, 2013.

[15] R. B. Perry, M. M. Madden, W. Torres-Pomales, and R. W. Butler, *The Simplified Aircraft-Based Paired Approach with the ALAS Alerting Algorithm*, Citeseer, 2013.

[16] S. Priess, "Analysis of an ADS-B in method for calculating the interval management paired approach collision safety limit," in *Proceedings of the 2017IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, pp. 1–8, IEEE, November 2017.

[17] N. A. Durkins, *Aircraft Type Designators*, Department of Transportation Air Traffic Organization Policy, FAA. JO 7360.1F, U.S, 2021.

[18] K. Leiden, S. Priess, P. Harrison, R. Stone, P. Strande, and M. Palmer, "Paired approach flight demonstration: planning and development activities," in *Proceedings of the 2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*, pp. 3G4-1–3G4-12, IEEE, Herndon, VA, USA, April 2018.

[19] H. Dai, D. I. Dongning, H. Qiao, and J. Cui, "Research progress and application prospects of aircraft wake detection technology," *Science and Technology Review*, vol. 31, no. 31, pp. 66–69, 2013.

[20] A. C. De Bruin, G. Hegen, P. B. Rohne, and P. R. Spalart, "Flow field survey in trailing vortex system behind a civil aircraft model at high lift," in *Proceedings of the Characterization & Modification of Wakes from Lifting Vehicles in Fluids, AGARD FDP Symposium*, pp. 25-1–25-12, Trondheim, Norway, May 1996.

[21] A. C. De Bruin and R. Oerlemans, *Flow Field Measurement Data Presentation and Analysis System for NLR Rake with 5-hole Probes*, NLR TR 96540 L, 1996.

[22] H. Vollmers, "Detection of vortices and quantitative evaluation of their main parameters from experimental velocity data," *Measurement Science and Technology*, vol. 12, no. 8, pp. 1199–1207, 2001.

[23] D. L. Ciffone, "Vortex interactions in multiple vortex wakes behind aircraft," *Journal of Aircraft*, vol. 14, no. 5, pp. 440–446, 1977.

[24] K. Huenecke, "Wake vortex investigations of transport aircraft," in *Proceedings of the 13th Applied Aerodynamics Conference*, p. 1773, San Diego, CA, USA, June 1995.

[25] K. Huenecke, "Structure of a transport aircraft-type near field wake," in *Proceedings of the Characterization & Modification of Wakes from Lifting Vehicles in Fluids*, pp. 5-1–5-9, AGARD FDP Symposium, Trondheim, Norway, April 1996.

[26] K. Huenecke, "Vortex wakes from large aircraft-A challenge for industrial research," in *Proceedings of the Fluids 2000 Conference and Exhibit*, p. 2216, Denver, CO, USA, June 2000.

[27] K. Hünecke, "Wake vortex control-a challenge for large transport aircraft," *Air & Space Europe*, vol. 3, no. 3-4, pp. 209–213, 2001.

[28] K. Huenecke, "The characterisation of transport aircraft vortex wakes," in *Proceedings of the 19th AIAA Applied Aerodynamics Conference*, p. 2427, A, June 2001.

[29] L. Jacquin, D. Fabre, P. Geffroy, and E. Coustols, *The properties of a transport aircraft wake in the extended near field-An experimental study*, 39th Aerospace Sciences Meeting and Exhibit, Reno, NV, 2001.

[30] C. Breitsamter, "Nachlaufwirbelsysteme großer transportflugzeuge-experimentelle charakterisierung und beeinflussung (wake-vortex systems of large transport aircraft—experimental characterization and manipulation)," Inaugural Thesis, Technische Universität München, München, Germany, 2007.

[31] V. J. Rossow and L. S. A. Branch, "Measurements in vortex wakes shed by conventional and modified subsonic aircraft," in *Proceedings of the Characterization & Modification of Wakes from Lifting Vehicles in Fluids*, pp. 26-1–26-9, AGARD FDP Symposium, Trondheim, Norway, November 1996.

[32] B. R. Cobleigh and J. Del Frate, *Water Tunnel Flow Visualization Study of a 4.4% Scale X-31 Forebody*, Citeseer, 1994.

[33] S. E. Sanders, O. R. Willis, N. H. Nahler, and E. Wrede, "Absolute fluorescence and absorption measurements over a dynamic range of 106with cavity-enhanced laser-induced fluorescence," *The Journal of Chemical Physics*, vol. 149, no. 1, Article ID 014201, 2018.

[34] J. Kang, Y. R. Wang, R. H. Li, and Y. Q. Chen, "Surface elemental microanalysis with submicron lateral resolution by the laser-ablation laser-induced fluorescence technique," *Optics Express*, vol. 26, no. 11, Article ID 14689, 2018.

[35] T. J. Myers, *Determination of Bragg Scatter in an Aircraft-Generated Wake Vortex System for Radar Detection*, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1997.

[36] H. Takami, *A Numerical experiment with Discrete Vortex Approximation with Reference to the Rolling up of a Vortex Sheet*, Stanford Univ. Report Sudaer, Stanford, CA, US, 1964.

[37] D. C. Lewellen and W. S. Lewellen, "The effects of aircraft wake dynamics on contrail development," *Journal of the Atmospheric Sciences*, vol. 58, no. 4, pp. 390–406, 2001.

[38] E. Coustols, L. Jacquin, and G. Schrauf, "Status of wake vortex alleviation in the frame work of European collaboration:

validation attempts using tests and CFD results," in *Proceedings of the European Conference on Computational Fluid Dynamics, ECCOMAS CFD*, Netherlands, January 2006.

[39] L. Mengda, C. Guixiang, Z. Zhaoshun, X. Chunxiao, and H. Weixi, "Large eddy simulation on the evolution and the fast-time prediction of aircraft wake vortices," *Journal of mechanics*, vol. 49, no. 6, pp. 1185–1200, 2017.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3334810, 14 pages
https://doi.org/10.1155/2021/3334810

Research Article

Short-Term Prediction of Traffic State for a Rural Road Applying Ensemble Learning Process

Arash Rasaizadi,[1] **Seyedehsan Seyedabrishami**,[1] **and Mohammad Saniee Abadeh**[2]

[1]*School of Civil and Environmental Engineering, Tarbiat Modares University, Tehran, Iran*
[2]*School of Electrical & Computer Engineering, Tarbiat Modares University, Tehran, Iran*

Correspondence should be addressed to Seyedehsan Seyedabrishami; seyedabrishami@modares.ac.ir

Received 8 June 2021; Accepted 17 December 2021; Published 29 December 2021

Academic Editor: Socrates Basbas

Short-term prediction of traffic variables aims at providing information for travelers before commencing their trips. In this paper, machine learning methods consisting of long short-term memory (LSTM), random forest (RF), support vector machine (SVM), and K-nearest neighbors (KNN) are employed to predict traffic state, categorized into A to C for segments of a rural road network. Since the temporal variation of rural road traffic is irregular, the performance of applied algorithms varies among different time intervals. To find the most precise prediction for each time interval for segments, several ensemble methods, including voting methods and ordinal logit (OL) model, are utilized to ensemble predictions of four machine learning algorithms. The Karaj-Chalus rural road traffic data was used as a case study to show how to implement it. As there are many influential features on traffic state, the genetic algorithm (GA) has been used to identify 25 of 32 features, which are the most influential on models' fitness. Results show that the OL model as an ensemble learning model outperforms machine learning models, and its accuracy is equal to 80.03 percent. The highest balanced accuracy achieved by OL for predicting traffic states A, B, and C is 89, 73.4, and 58.5 percent, respectively.

1. Introduction

Sustainable transportation networks need to use data obtained from intelligent transportation systems (ITS) to relieve traffic congestion and its consequences, such as air and noise pollution and wasting energy and time. Intelligent traffic congestion alleviation is a vital element of smart mobility and smart transportation systems [1]. One of the intelligent transportation systems is the advanced traveler information system (AITS). AITSs provide useful information about the current or future traffic conditions to travelers and transportation agencies [2]. These systems' effectiveness is more when predicting the future state of the transportation network and letting users have better plans for their next trips [3]. A group of travelers who plan to travel during traffic peak hours will more likely postpone or cancel their trips. These changes lead to more balanced distributed trips over time and a more sustainable transportation network. Traffic volume and average speed are well-known continuous traffic variables that can be predicted [4, 5]. Many users are unable to understand the performance condition of the transportation network by knowing these variables. The traffic volume to capacity ratio and average speed to free-flow speed ratio are more informative and meaningful for users [6]. Instead of predicting traffic volume and speed, we can predict the traffic state as a nominal traffic variable. This variable is determined regarding the volume to capacity ratio and the speed to free-flow speed ratio.

Another critical point is employing appropriate models that are compatible with the nominal nature of the traffic state. Predictive methods are diverse, and there is no superior model for every prediction problem [7]. Generally, predictive techniques are divided into naïve, time series, and machine learning [8].

Naïve methods are simple with short computational time. These methods do not react to dynamic changes and

usually are used as a benchmark [8]. Time-series methods (also known as parametric or statistical methods) have a well-established theoretical background and show the importance and effect of independent variables on the dependent variable by estimating coefficients and t states [9]. However, one of these models' deficiencies is the inability to depict nonlinear relationships because of the assumption's limitations [10]. By increasing the volume of the dataset, these methods need more computational processing power. Also, these methods concentrate on means and miss the extremes [11]. The family of time-series methods includes autoregression (AR), moving average (MA), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and seasonal autoregressive integrated moving average (SARIMA) [12]. Alghamdi et al. [13] leverage ARIMA-based modeling to forecast traffic congestion. Analyzing nonstationary and nonnormally distributed traffic data by ARIMA achieves appropriate performance with a confidence level of 95%. Ding et al. [14] forecast subway ridership by ARIMA-GHARCH. The proposed model has a more accurate prediction compared to the ARIMA-only model.

Machine learning methods (also known as nonparametric methods) are capable of mapping nonlinear relationships. These methods are more suitable for analyzing big data and having no or fewer assumptions [15]. The main disadvantages of machine learning methods are lack of interpretability, needing many observations to train, and working like a black box [16]. Learning models based on neural networks such as long short-term memory (LSTM) [17], SVM [18], K-nearest neighbor (KNN) [19], and random forest (RF) [20] are some machine learning methods to predict traffic variables. Du et al. [21] predict traffic passenger flow for urban areas and propose a deep irregular convolutional residual LSTM network (DST-ICRL). By using both short-term and long-term historical data, the proposed method outperforms traditional machine learning methods. To predict traffic flow, Kang et al. [22] use LSTM recurrent neural network. They conclude that occupancy, speed, and downstream and upstream traffic information as predictor variables can enhance prediction accuracy. A spatiotemporal correlative K-nearest neighbor is proposed by Cai et al. [23]. Gaussian weighted Euclidean finds the nearest neighbors. Also, considering the relationship between road segments improves model performance. Using the RF model, Liu and Wu [24] predict traffic congestion. The weather conditions, time, special road conditions, road quality, and holiday are used in the predictive RF model. Li et al. [25] propose a combined WT-FCBF-LSTM (wavelet transform, fast correlation-based filter, and long short-term memory) model to predict passenger demand by hybrid ridesharing service models. The proposed model has better performance in terms of accuracy compared to single LSTM and single WT-LSTM models.

Diverse prediction methods have different advantages and disadvantages. Ensemble learning is a process that receives model predictions as input and makes a unique final prediction. Moretti et al. [26] predict traffic flow by using a statistical and neural network bagging ensemble hybrid model, which outperforms the prediction of input methods. Also, Yang et al. [27] show that the gradient boosting decision trees (GBDT) bring more prediction accuracy than the SVM and backpropagation neural networks for their case study.

In the current study, the hourly traffic state consisting of light, semiheavy, and heavy traffic is predicted for one section of Karaj-Chalus, a rural road in Iran. Many features related to traffic state variation are extracted and used as predictor features. One of the essential parts of modeling is feature selection. Features could be selected by try and error, but there are some systematic methods. In this study, by using the genetic algorithm (GA), influential features are selected systematically. The next step is to employ machine learning methods. Several machine learning methods, including LSTM as a deep learning approach for time-series prediction, KNN, SVM, and RF, are trained to predict traffic states. Finally, ensemble methods, including OL and four voting methods, convert LSTM, KNN, SVM, and RF predictions to one final prediction. It is expected that ensemble methods provide more accurate predictions compared to the initial predictive methods.

Compared to traffic volume and speed, nominal traffic state is more informative for travelers. It can be shown in traffic maps by the easily understandable colors. Travelers decide about departure time and trip route by information obtained from advanced traveler information systems. Also, transportation agencies can benefit from accurate traffic state predictions. It is vital to provide accurate predictions at any time. It motivates us to propose an ensemble learning process that is expected to have more stable performance in terms of prediction accuracy than single models.

The first contribution of this paper is related to the data. We add new features related to date, solar and lunar calendars, weather conditions, and road blockage. Also, we predict traffic state as a nominal variable, investigate rural traffic data with nonroutine trips, and use Iran's traffic data as a developing country. Second, the feature selection is made by GA and two datasets train models; the first one includes selected features by GA, and the second one contains all features. The accuracy of the models for each dataset is calculated and compared. The ensemble learning process by OL and voting methods is another contribution of the current study. The performance of single machine learning and ensemble learning algorithms for hourly traffic state prediction is evaluated in different evaluation metrics.

2. Data

Karaj-Chalus is a rural road in the north of Iran, a part of a route from Tehran, the capital of Iran, to the seaside. The length of this road is 170 kilometers. In addition, there are three parallel roads with this road but with different lengths. Many trips to Chalus are recreational and nonmandatory. These nonroutine trips make the prediction more difficult because finding traffic patterns is not simply compared to routine trips. Figure 1 shows the map of Karaj-Chalus road.

The purpose of this paper is to predict the hourly traffic state. Traffic state is a more informative variable for travelers

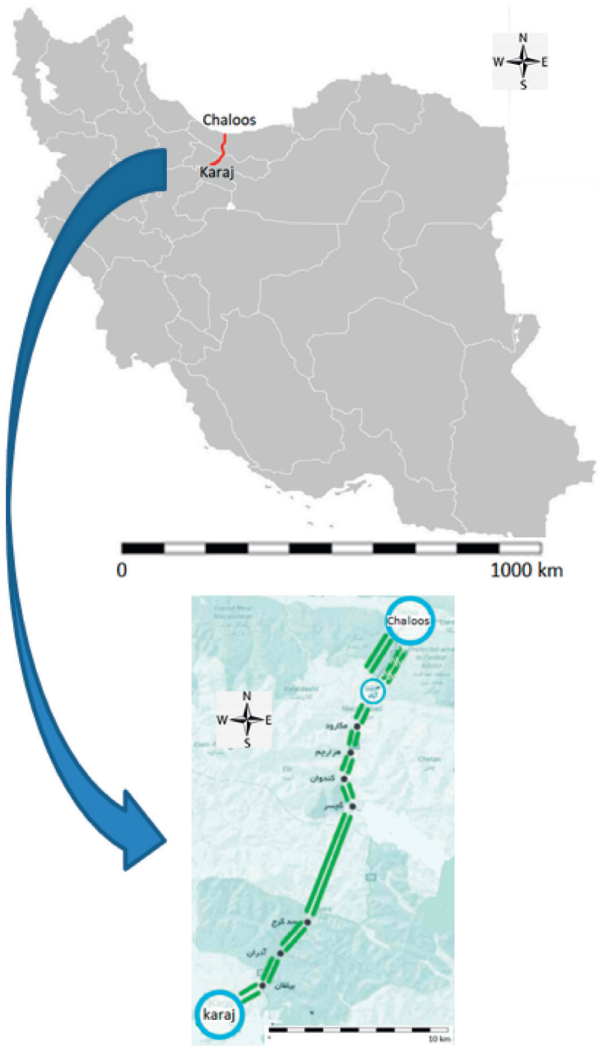


Figure 1: Map of Karaj-Chalus road in Iran.

who do not know other characteristics of the road. Loop detectors in one section of this road collect hourly traffic volume and hourly average speed. By calculating the hourly traffic volume over the hourly capacity ratio and hourly average speed over the free-flow speed ratio, the hourly traffic state is determined based on Table 1. A, B, and C represent light, semiheavy, and heavy traffic, respectively. This type of traffic state definition is provided by Iran Road Maintenance and Transportation Organization (RMTO, http://www.rmto.ir/).

The raw data only has hourly traffic state, hourly traffic volume, hourly average speed, and date. One of the essential steps before training models is extracting effective features. Traffic patterns of nonroutine trips are affected by holidays and different types of holidays have different effects. Holidays in Iran are based on solar and lunar calendars. Since solar and lunar calendars are not fixed together, both of them are considered simultaneously. So, several features related to holidays and their types are defined based on lunar and solar calendars. Police often blocked this road in each direction or parallel roads for traffic management at peak hours. Therefore,

Table 1: Determining hourly traffic state.

S/S_f*	V/C*					
	Under 0.1	0.1–0.3	0.3–0.5	0.5–0.7	0.7–0.9	Over 0.9
Over 0.95	A	A	A	B	B	C
0.8–0.95	A	A	B	B	B	C
0.6–0.8	A	B	B	B	C	C
0.45–0.6	B	B	B	C	C	C
Under 0.45	C	C	C	C	C	C

*V and C are hourly traffic volume and the hourly capacity and V/C is their ratio. S and Sf are hourly average speed and the free-flow speed and S/Sf is their ratio. More V/C equals fewer S/Sf and more heavy traffic. Less V/C equals more S/Sf and less congested traffic. Thresholds are defined based on observed traffic patterns and RMTO experts' judgment. For example, S/Sf under 0.3 is rarely observed, so it is not necessary to consider it as a threshold.

blockage of the road, blockage of opposite direction, and blockage of parallel roads are added to the dataset as features. Table 2 shows all the features, which are extracted in this study.

This data is collected for 17 months, from March 2017 to August 2018. The first 12 months of the dataset are used for training single models (train dataset 1), the OL model is calibrated by the next three months (train dataset 2), and the last two months (test dataset) is used to test the predictions of single and ensemble methods. Also, two months of train dataset 1 is used for cross-validation to tune the models' parameters and evaluate the performance of models in a robust manner. For this purpose, another method is using Monte-Carlo simulation [28]. The total number of observations is 11353. Table 3 shows the frequency of traffic states for each part of the dataset. Pie charts in Figure 2 show the characteristics of candidate features.

All features are used to train the models, but some of these features may have less effect on the models' prediction power or even have a negative effect on prediction. In this study, to select effective predictors systematically and include them in models, GA is used. The following procedure is employed for feature selection [29]:

Step 1: define population size (P) for each generation, mutation probability (pm), and stopping criteria.

Step 2: randomly generate an initial population of chromosomes.

Step 3: repeat until the stopping criterion is met:

- For each chromosome, do
  Tune and train the classifier model and compute each chromosome's fitness
  End.
  For each reproduction 1 to P/2, do
  Select two chromosomes based on fitness.
  Crossover: randomly select a locus and exchange genes (a mechanism to form new genes) on either side of the locus to produce two-child chromosomes with mixed genes.
  Mutate the child chromosomes with probability pm.
  End.

Table 2: Description of extracted candidate features to predict traffic state.

Feature name	Description
Season	Including spring, summer, fall, and winter
Solar month	Including 12 solar months
Lunar month	Including 12 lunar months
Day of a solar month	Including 29–31 days of a solar month
Day of a lunar month	Including 29–30 days of a lunar month
Time of day	Including 24 hours a day
Six hours before holidays	Equal to 1 if it is 1 to 6 hours before holidays
Six hours after holidays	Equal to 1 if it is 1 to 6 hours after holidays
Day or night	Including day and night
Number of holidays	The number of sequential holidays
Holidays	Including 1 for holidays and 0 for other days
Holiday type	Type of holidays
Holiday in three days later	Equal to 1 if three days later is a holiday
Type of holidays in three days later	Including the holiday type of three days later if it is a holiday; otherwise, it equals 0
Holiday in three days ago	Equal to 1 if three days ago is a holiday
Type of holidays in three days ago	Including the holiday type of three days ago if it is a holiday; otherwise, it equals 0
Holiday in two days later	Equal to 1 if two days later is a holiday
Type of holidays in two days later	Including the holiday type of two days later if it is a holiday; otherwise, it equals 0
Holiday in two days ago	Equal to 1 if two days ago is a holiday
Type of holidays in two days ago	Including the holiday type of two days ago if it is a holiday; otherwise, it equals 0
Holiday in a day later	Equal to 1 if a day later is a holiday
Type of holidays in a day later	Including the holiday type of a day later if it is a holiday; otherwise, it equals 0
Holiday in a day ago	Equal to 1 if a day ago is a holiday
Type of holidays in a day ago	Including the holiday type of a day ago if it is a holiday; otherwise, it equals 0
Weather condition	Including sunny, rainy, and snowy
Blockage	Blockage of the road by police
Blockage of the opposite direction	Blockage of the opposite direction by police
Blockage of parallel paths	Blockage of parallel paths by police

Table 3: Frequency of traffic states in observations.

Traffic state	Train dataset 1		Train dataset 2		Test dataset	
	Frequency	Frequency percentage	Frequency	Frequency percentage	Frequency	Frequency percentage
Light (A)	3708	46.38	741	31.55	170	16.88
Semiheavy (B)	3822	47.80	1365	58.11	693	68.82
Heavy (C)	465	5.82	243	10.34	144	14.30

Chromosomes, which consist of genes, are binary vectors with 1 representing a feature's presence and 0 its absence. The population is a set of chromosomes (solutions). In the reproduction algorithm, the two-parent chromosomes are split at a random position, and the head of one chromosome is combined with the tail of the other chromosome [29]. The prediction accuracy (fitness) of an internal decision tree is the objective function.

This procedure is implemented in the R software. Among all features, seven features are not qualified by GA. These features are as follows:

(1) Type of holidays in a day later.

(2) Holiday in a day later.

(3) Holiday in a day ago.

(4) Holiday in three days later.

(5) Blockage.

(6) Blockage of the opposite direction.

(7) Blockage of parallel paths.

Models are trained by selected features by GA and all of the features.

3. Methodology

3.1. Long Short-Term Memory. Recurrent neural networks (RNNs) are deep artificial neural network (ANN) models that keep information in memory. These models consider the dependency between sequential observations. The chief defect of these models is that they only consider short-term dependencies because the gradient of loss function declines exponentially over time. LSTM is a kind of RNN that can handle the long-term dependencies alongside short-term dependencies [30, 31].

The LSTM structure consists of four gates (neural network layers), forget, remember, learn, and output gates. The LSTM model's inputs are long-term memory, short-term memory, and training example (new data). The long-term input goes into the forget gate (1), which decides to forget irrelevant parts. The short-term and training example inputs

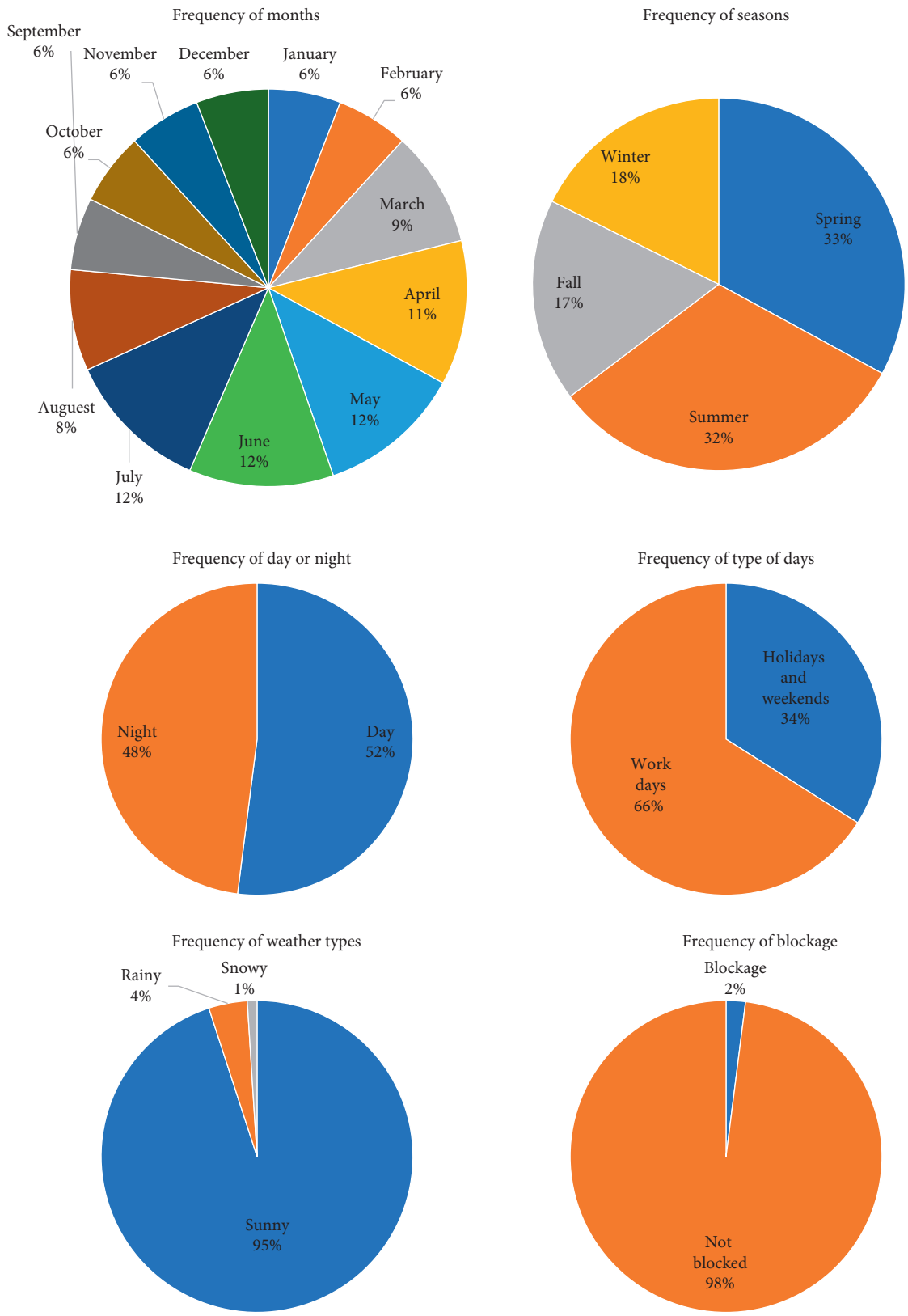


Figure 2: Characteristics of candidate features.

go into the learn gate (2), which determines what inputs are to be learned. Passed information (consisting of short- and long-term memories) from forget and learn gates goes into the remember gate (3), producing new long-term memories for the output gate. Finally, the output gate (4) updates short-term memories and the model's final output [31]. The equations of gates in LSTM are

$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right), \tag{1}$$

$$l_t = \tanh\left(w_n\left[h_{t-1}, x_t\right] + b_n\right), \tag{2}$$

$$r_t = l_t + f_t, \tag{3}$$

$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right), \tag{4}$$

$$\sigma(\mathrm{t}) = \frac{1}{1 + e^t}. \tag{5}$$

f_t, l_t, r_t, and o_t are factors of forget, learn, remember, and output gates, respectively. σ is the sigmoid function (5). w_x is the weight for the gate(x) neurons. h_{t-1} is the output of the previous LSTM block. x_t is the input at the current timestamp and b_x is the bias for the gate(x). Figure 3 shows the architecture of the LSTM network.

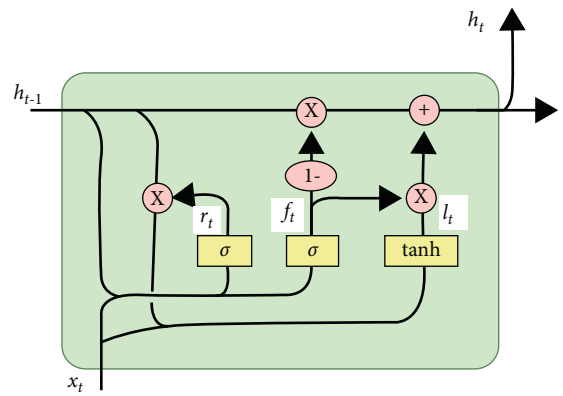


Figure 3: Architecture of an LSTM network [32].

3.2. Support Vector Machine. SVM is a supervised machine learning classifier used for classification and regression (SVR) problems. This model finds a hyperplane in an N-dimensional space to classify the data points distinctly. The model finds a hyperplane with the maximum distance between data points of classes (support vectors). The loss function of SVM to maximize the margin is hinge loss. Future data can be classified based on their position relative to that hyperplane [33].

In many real situations, the data is not linearly separable. Applying a transformation by the kernel function is essential before classification. This study uses the radial basis function (RBF) kernel function among different kernel functions. The formulation of RBF function is as (6) [34]:

$$\mathrm{K}\left(\mathrm{X_i}, \mathrm{X}\right) = \exp\left(\frac{\mathrm{X_i} - \mathrm{X_j}^2}{2\sigma^2}\right), \tag{6}$$

where σ is a free parameter to be calibrated. $X_i - X_j{}^2$ is the squared Euclidean distance between the two feature vectors X_i and X_j.

3.3. K-Nearest Neighbor. The KNN model is a supervised machine learning algorithm for both classification and regression problems. The main idea of KNN is to find a predefined number of training samples (K) closest in the distance to the new point and predicts the class by voting. This algorithm can be summarized in 4 steps [23].

Step 1: store the training samples in an array of data points.

Step 2: calculate the distance of training samples and new data point p.

Step 3: find the K smallest distance obtained.

Step 4: return the majority class of K smallest distance.

Euclidean, Manhattan, and Minkowski are well-known distance functions. This paper used Euclidean distance to calculate the distance between data points, X_i and X_j:

$$\begin{aligned} d\left(X_i, X_j\right) &= \sqrt{\left(X_{i1} - X_{j1}\right)^2 + \left(X_{i2} - X_{j2}\right)^2 + \cdots + \left(X_{in} - X_{jn}\right)^2}, \\ &= \sqrt{\sum_{i=1}^{n}\left(X_{in} - X_{jn}\right)^2}. \end{aligned} \tag{7}$$

3.4. Random Forest. RF is a supervised learning algorithm that can be used for both classification and regression. It consists of many individual decision trees that spit out a class prediction, and the class with the most votes becomes our model's prediction. The following steps show how this algorithm works [35]:

Step 1: start with the select random samples from the training dataset.

Step 2: construct a decision tree for every sample.

Step 3: get the prediction result from every decision tree.

Step 4: perform voting for every predicted result.

Step 5: the most voted prediction result is the final prediction result.

Decision trees start with a node and branch to another node. This paper uses the entropy formula to determine how the dataset branches from each node. Equation (8) presents the entropy formula [36].

$$\text{Entropy} = \sum_{i=1}^{c} - p_i * \log_2\left(pi\right), \tag{8}$$

where p_i is the relative frequency of class i, i is the index of classes, and c is the total number of classes.

3.5. Ensemble Learning. At this step, ensemble learning methods put predictions of introduced methods together to provide one unique final prediction. The final prediction is expected to have higher accuracy than the accuracy of LSTM, SVM, KNN, and RF.

Four different voting methods are defined as follows:

Voting to a better state: predictions are the majority vote of contributing models. If majorities are equal, priority is A, B, and C, respectively.

Voting to a worse state: predictions are the majority vote of contributing models. If majorities are equal, priority is C, B, and A, respectively.

Best state: it selects A if at least one model predicts A, else it selects B if at least one predicts B; otherwise, it selects C.

Worst state: it selects C if at least one model predicts C, else it selects B if at least one predicts B; otherwise, it selects A.

Another method is OL, which is a statistical method. In this method, input models' importance for predicting each traffic state is determined by estimating coefficients and t-state. Let us define s^*_{qi} is a linear function consisting of a vector of input variables x_{qi}, corresponding coefficients γ, and random term η_{qi}. q is the index of hour, i is the index of traffic state, and δ_{AB} and δ_{BC} are thresholds [37].

$$s^*_{qi} = \gamma x_{qi} + \eta_{qi}. \tag{9}$$

The final output is A if $s^*_{qi} \leq \delta_{AB}$, is B if $\delta_{AB} < s^*_{qi} \leq \delta_{BC}$, and is C if $\delta_{BC} < s^*_{qi}$.

Model parameters are estimated by maximizing the likelihood function.

$$\log L = \sum_{i=1}^{3} M_{qi} \log\left[\Pr_{qi}\right], \tag{10}$$

where M_{qi} is equal to 1 if state i occurred in hour q, else zero, and $\Pr_{qi}$ is occurrence probability of state i in hour q [37].

Figure 4 shows the proposed ensemble learning process.

All of the models are implemented in the R software.

4. Results and Discussion

LSTM, SVM, KNN, and RF models are trained by train dataset 1. To tune the parameters of models, different values are set for them. The final parameters are selected in terms of the accuracy of predictions on the rest of the dataset. These parameters include K in KNN, the number of trees to grow (NT), and the number of variables randomly sampled as candidates at each split (NV) in the RF model and cost (C) in the SVM model. A summary of parameters tuning of models is presented in Table 4.

After training models by optimum parameters, the accuracy of predictions on train dataset 1 and train dataset 2+test datasets are calculated and presented in Table 5.

According to Table 5, RF, SVM, KNN, and LSTM models can be sorted in terms of accuracy. Results indicated that the RF model trained by the GA features outperforms other models with 78.66% accuracy. Also, SVM and KNN have similar performances, and LSTM predictions have less accuracy than the other models. Using GA features increases the efficiency of LSTM, KNN, and RF models. Reducing data dimension and shortening computational time without eliminating any useful information are other advantages of feature selection by GA.

Figure 5 shows the accuracy changes during the time. The horizontal axis represents weeks in the train 2+test dataset, and the vertical axis represents accuracies in percent. Also, Figure 6 shows the prediction accuracy of models for three random days. Based on Figures 5 and 6, there is no most accurate model for all the times due to the temporal variation of accuracies. This finding emphasizes using ensemble methods to provide unique predictions with the highest possible accuracy.

At the next step, OL is calibrated by using predictions of contributing models for train dataset 2. The performance of ensemble methods, including voting algorithms and OL and single models, is evaluated for the test dataset. The results are presented in Table 6.

Table 6 shows that the OL model outperforms its input with 81.03% prediction accuracy. For every period, the OL model can detect the more accurate single model and put more value on that model's prediction. After OL, the worst state voting algorithm provides 78.35% of accuracy. It means that single models have more tendencies to predict light state and sometimes miss heavy state.

It is essential to evaluate models for predicting each traffic state. For this purpose, among diverse evaluation metrics, F-measure (F_1) and balanced accuracy (because of an unbalanced distribution of observations) are calculated by using confusion matrices (Table 7) and equations 11 and 12 (see Akosa, 2017, Labatut and Cherifi, 2012, and Tharwat, 2018, about calculating recall, precision, and specificity).

$$\text{Balanced accuracy} = \frac{(\text{Re call} + \text{Specificity})}{2}, \tag{11}$$

$$F_1 = \frac{2(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}}. \tag{12}$$

These metrics are calculated for single and ensemble methods and presented in Table 8.

Based on Table 8, the OL predicts all states more accurately than voting and single machine learning methods. This model predicts states A, B, and C with balanced accuracy equal to 89, 73.4, and 58.5 percent, respectively. Also, OL's F_1s equal 0.813 for state A, 0.872 for state B, and 0.292 for state C and all of them are the

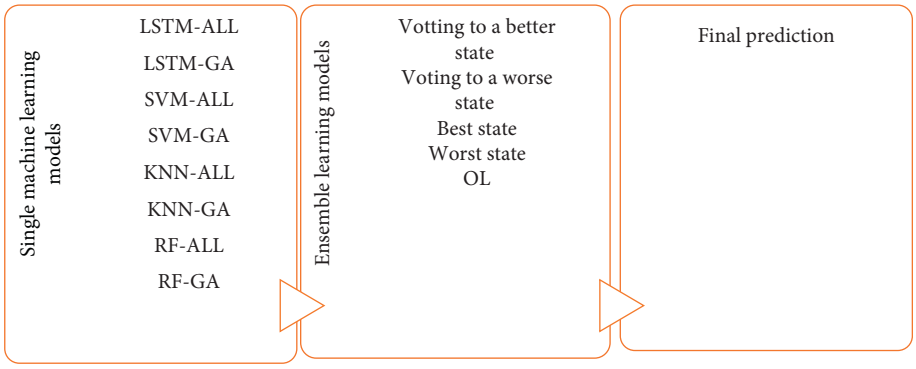


Figure 4: Proposed ensemble learning process.

Table 4: Optimum values for parameters of models.

Parameter	Model	The optimum value by using all features	The optimum value by using selected features by GA
K	KNN	26	37
NT	RF	162	109
NV	RF	9	8
C	SVM	5	3

Table 5: Accuracy of predictions of single models.

Model	Train 1-all features	Train 2+test-all features	Train 1-GA features	Train 2+test-GA features
LSTM	76.65	70.77	75.93	70.89
SVM	82.09	76.25	80.60	75.48
KNN	80.10	74.94	79.15	75.41
RF	97.94	78.31	98.09	78.66

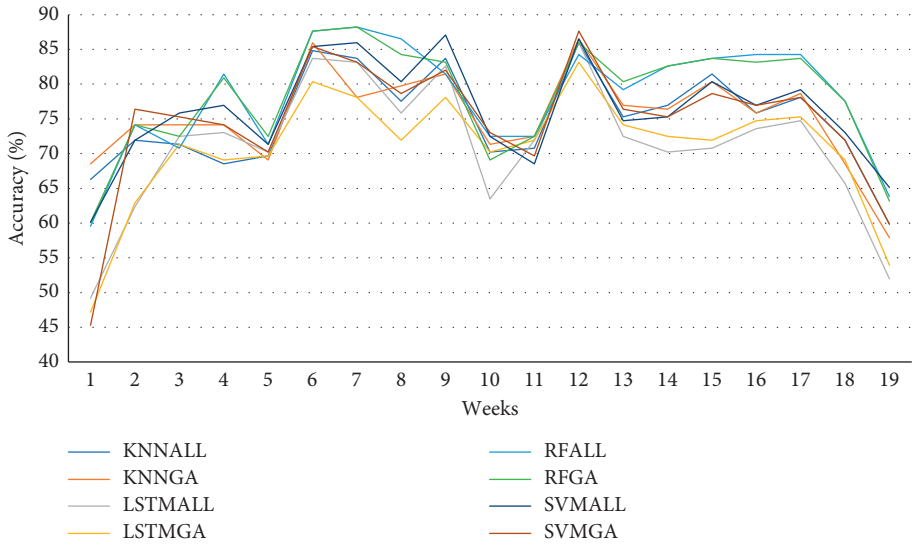


Figure 5: Accuracy changes of single models during the time.

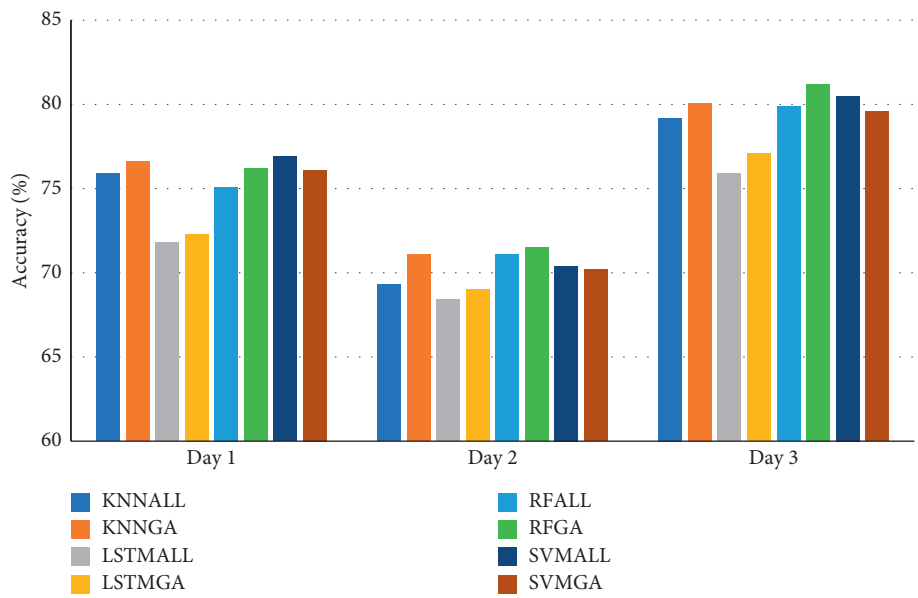


Figure 6: The prediction accuracy of models for three random days.

Table 6: Accuracy of predictions of ensemble and single models for test datasets.

Model	Test-all features	Test-GA features
LSTM	68.92	70.71
SVM	75.37	74.38
KNN	74.18	73.58
RF	76.96	76.86
Voting to a better state	75.17	
Voting to a worse state	75.67	
Best state	67.63	
Worst state	78.35	
OL	81.03	

Table 7: Confusion matrices of predictions for the test dataset.

True prediction	A	B	C
Model		LSTM-all	
A	151	151	3
B	19	542	140
C	0	0	1
Model		SVM-all	
A	136	81	0
B	34	610	131
C	0	2	13
Model		KNN-all	
A	144	97	1
B	26	585	125
C	0	11	18
Model		RF-all	
A	150	78	0
B	20	614	133
C	0	1	11

Table 7: Continued.

True prediction	A	B	C
Model		Voting to a better state	
A	149	86	1
B	21	607	142
C	0	0	1
Model		Best state	
A	162	175	3
B	8	518	140
C	0	0	1
Model		OL	
A	141	36	0
B	29	649	118
C	0	8	26
Model		LSTM-GA	
A	137	119	1
B	33	574	142
C	0	0	1
Model		SVM-GA	
A	143	88	1
B	27	605	142
C	0	0	1
Model		KNN-GA	
A	144	90	1
B	26	594	140
C	0	9	3
Model		RF-GA	
A	151	76	0
B	19	613	134
C	0	4	10
Model		Voting to a worse state	
A	142	76	1
B	28	617	140
C	0	0	3
Model		Worst state	
A	116	26	0
B	54	645	116
C	0	22	28

Table 8: Evaluation metrics for predicting each traffic state.

Model	State	Precision	Recall	Specificity	Balanced accuracy	F1
LSTM-all	A	0.495	0.888	0.779	0.834	0.636
	B	0.773	0.782	0.489	0.635	0.778
	C	1	0.007	1	0.503	0.014
LSTM-GA	A	0.533	0.806	0.827	0.817	0.642
	B	0.766	0.828	0.441	0.635	0.796
	C	1	0.007	1	0.503	0.014
SVM-all	A	0.627	0.8	0.885	0.842	0.703
	B	0.787	0.88	0.475	0.677	0.831
	C	0.867	0.09	0.997	0.544	0.164

Table 8: Continued.

Model	State	Precision	Recall	Specificity	Balanced accuracy	F1
SVM-GA	A	0.616	0.841	0.872	0.857	0.711
	B	0.782	0.873	0.46	0.667	0.825
	C	1	0.007	1	0.503	0.014
KNN-all	A	0.616	0.841	0.872	0.857	0.711
	B	0.782	0.873	0.46	0.667	0.825
	C	1	0.007	1	0.503	0.014
KNN-GA	A	0.613	0.847	0.868	0.857	0.711
	B	0.782	0.857	0.47	0.663	0.818
	C	0.25	0.021	0.988	0.504	0.038
RF-all	A	0.658	0.882	0.889	0.886	0.754
	B	0.801	0.886	0.513	0.699	0.841
	C	0.917	0.076	0.999	0.538	0.141
RF-GA	A	0.665	0.888	0.891	0.889	0.761
	B	0.8	0.885	0.513	0.699	0.84
	C	0.714	0.069	0.995	0.532	0.127
Voting to a better state	A	0.631	0.876	0.875	0.876	0.734
	B	0.788	0.876	0.479	0.678	0.83
	C	1	0.007	1	0.503	0.014
Voting to a worse state	A	0.648	0.835	0.89	0.862	0.73
	B	0.786	0.89	0.463	0.677	0.835
	C	1	0.021	1	0.51	0.041
Best state	A	0.476	0.953	0.745	0.849	0.635
	B	0.778	0.747	0.524	0.636	0.762
	C	1	0.007	1	0.503	0.014
Worst state	A	0.817	0.682	0.963	0.823	0.744
	B	0.791	0.931	0.459	0.695	0.855
	C	0.56	0.164	0.972	0.583	0.289
OL	A	0.797	0.829	0.949	0.89	0.813
	B	0.815	0.937	0.532	0.734	0.872
	C	0.765	0.181	0.99	0.585	0.292

highest achieved F_1 for each traffic state. Table 7 shows that using the OL increases the balanced accuracy of predicting traffic states A, B, and C, about 0.1%, 3.5%, and 4.1%, respectively, compared to the highest accuracy achieved by the models it puts together.

OL coefficients help to find the importance of predicting each traffic state by each model. Table 9 shows the results of the OL model. Predictions of models converted to binary (dummy) variables to be used in the OL model. Values in parentheses show t-state.

Negative coefficients decrease s^*_{qi}. It means that they increase the probability of states A and B compared to state C, as a base traffic state. For example, predicting traffic state A by KNN-all decreases s^*_{qi} by 1.57 units. This decrease leads to an increase in the occurrence probability of state A compared to the traffic state C. T-state under 1.56 shows statistically insignificant variables at the 90% level of significance. For example, LSTM predictions are statistically insignificant in predicting traffic states.

Theoretically, the proposed ensemble learning process has no prediction time horizon limitation, but the accuracy of prediction models decreases as time passes. The prediction horizon is different in previous studies. Some previous studies suggest 6 months to have accurate prediction [38], but it completely depends on the employed model and data. Also, Figure 5 shows that the prediction accuracy of single models decreases dramatically after 17 weeks. Finally, 6-month prediction time horizon seems to be suitable based on the literature and Figure 5.

Finally, predicted traffic states could be informed to travelers and transportation operators via advanced traveler information systems. Travelers will have more insights for choosing their departure times and routes to destinations. Also, using these predictions, system operators are better prepared to deal with unsuitable traffic conditions, and they may implement policies such as access restrictions or increasing the number of route lanes on the schedule to avoid high congestion.

Table 9: Results of the OL model.

Traffic states	KNN-all	KNN-GA	LSTM-all	LSTM-GA	RF-all	RF-GA	SVM-all	SVM-GA
State A	−1.57 (−3.46)	−2.88 (−5.00)	−0.18 (−0.46)	−0.64 (−1.15)	−2.53 (−30.1)	−3.82 (−4.16)	−3.17 (−7.91)	2.36 (4.09)
State B	−1.16 (−2.94)	−1.79 (−3.41)	−0.13 (−0.44)	−0.64 (−1.27)	−1.37 (−1.75)	−2.66 (−3.05)	−2.12 (−7.00)	1.95 (3.99)
State C (base)	—	—	—	—	—	—	—	—
Other parameters								
Intercept	8.47 (10.93)	Threshold 1	−2.01 (−5.02)	Threshold 2	2.52 (6.56)	Log-likelihood	−1352.3	

5. Conclusion

Short-term traffic state prediction is a tool in the advanced traveler information system that aims to bring a more sustainable and more reliable transportation network. By predicting the near future of transportation network performance, travelers and system operators are more ready to face congested traffic or avoid getting stuck in traffic. This paper predicts the nominal practical traffic state that is more understandable for travelers. Many features are extracted in the preprocessing step related to solar and lunar calendars, weather conditions, and blockages. Feature selection is made by GA systematically. Then machine learning models consisting of LSTM, KNN, SVM, and RF models are trained using the GA selected features and all features. Ensemble methods, including four voting methods and the OL model, use all predictions and predict one final prediction to inform the road passengers and transportation agencies. The final results show that OL obtains the highest accuracy among machine learning and ensemble learning algorithms, which equals 81.03%. The highest accuracy of single machine learning methods is 76.96%, achieved by RF. The feature selection by GA maintains the accuracy of predictions and increases the accuracy of some models. Regarding F_1 and balanced accuracy, traffic states A, B, and C are predicted more accurately by the OL model in the ensemble learning process. This model provides interpretable coefficients, which can be used to show the importance of models prediction.

For future studies, using and comparing other ensemble learning methods such as gradient boosting decision trees (GBDT) and neural network bagging ensemble hybrid model is proposed.

Data Availability

The traffic time-series data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Acknowledgments

The authors would like to acknowledge the Road Maintenance and Transportation Organization (RMTO) for supporting this research by providing suburban traffic data of Iran.

References

[1] S. Majumdar, M. M. Subhani, B. Roullier, A. Anjum, and R. Zhu, "Congestion prediction for smart sustainable cities using IoT and machine learning approaches," *Sustainable Cities and Society*, vol. 64, Article ID 102500, 2021.

[2] Y. E. Mokaddem and F. Jawab, "Researches and applications of intelligent transportation systems in urban area: systematic literature review," *ARPN J. Eng. Appl. Sci*, vol. 14, no. 3, pp. 639–652, 2019.

[3] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the 2015 IEEE International Conference on Smart city/SocialCom/SustainCom (SmartCity)*, December 2015.

[4] S. Shahriari, M. Ghasri, S. A. Sisson, and T. Rashidi, "Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction," *Transportmetrica: Transportation Science*, vol. 16, no. 6, pp. 1–22, 2020.

[5] L. Zheng, H. Huang, C. Zhu, and K. Zhang, "A tensor-based K-nearest neighbors method for traffic speed prediction under data missing," *Transportation Business: Transport Dynamics*, vol. 8, no. 1, pp. 182–199, 2020.

[6] M. Zahid, Y. Chen, A. Jamal, and M. Q. Memon, "Short term traffic state prediction via hyperparameter optimization based classifiers," *Sensors*, vol. 20, no. 3, 685 pages, 2020.

[7] T. P. Carvalho, F. A. A. M. N. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. S. Alcalá, "A systematic literature review of machine learning methods applied to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, Article ID 106024, 2019.

[8] G. Chang, Y. Zhang, D. Yao, and Y. Yue, "A summary of short-term traffic flow forecasting methods," in *Proceedings of the ICCTP 2011: Towards Sustainable Transportation Systems*, pp. 1696–1707, Nanjing, China, July 2011.

[9] P. M. Maçaira, A. M. T. Thome, F. L. C. Oliveira, and A. L. C. Ferrer, "Time series analysis with explanatory variables: a systematic literature review," *Environmental Modelling & Software*, vol. 107, pp. 199–209, 2018.

[10] J. Fan and Q. Yao, "Nonlinear Time Series," *Nonparametric and Parametric Methods*, Springer Science & Business Media, New York, NY. USA, 2008.

[11] E. Bowitz, "Disability benefits, replacement ratios and the labour market. A time series approach," *Applied Economics*, vol. 29, no. 7, pp. 913–923, 1997.

[12] B. Kedem and K. Fokianos, *Regression models for time series analysis*, Vol. 488, John Wiley & Sons, , Hoboken, New Jersey. USA, 2005.

[13] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting traffic congestion using ARIMA modeling," in *Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, June 2019.

[14] C. Ding and J. Duan, Y. Zhang, X. Wu, and G. Yu, Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1054–1064, 2017.

[15] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.

[16] E. Alpaydin, *Introduction to Machine Learning*, MIT press, Cambridge, MA, USA, 2020.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[19] L. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[20] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[21] B. Du, H. Peng, S. Wang et al., "Deep irregular convolutional residual lstm for urban traffic passenger flows prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[22] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, October 2017.

[23] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 21–34, 2016.

[24] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," in *Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, December 2017.

[25] X. Li, Y. Zhang, M. Du, and J. Yang, "The forecasting of passenger demand under hybrid ridesharing service modes: a combined model based on WT-FCBF-LSTM," *Sustainable Cities and Society*, vol. 62, p. 102419, 2020.

[26] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.

[27] S. Yang, J. Wu, Y. Du, Y. He, and X. Chen, "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *Journal of Sensors*, vol. 2017, Article ID 7074143, 2017.

[28] N. D. Bokde, Z. M. Yaseen, and G. B. Andersen, "ForecastTB-an R package as a test-bench for time series forecasting-application of wind speed and solar radiation modeling," *Energies*, vol. 13, no. 10, 2578 pages, 2020.

[29] M. Kuhn and K. Johnson, *Applied predictive modeling*, Vol. 26, Springer, , New York, NY. USA, 2013.

[30] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, Wiley, Hoboken, New Jersey, United States, 2001.

[31] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, Article ID 132306, 2020.

[32] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "LSTM-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[33] S. Mokhtarimousavi, J. C. Anderson, A. Azizinamini, and M. Hadi, "Improved support vector machine models for work zone crash injury severity prediction and analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 11, pp. 680–692, 2019.

[34] S. Han, C. Qubo, and H. Meng, "Parameter selection in SVM with RBF kernel function," in *World Automation Congress*IEEE, 2012.

[35] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[36] M. Yin, D. Yao, J. Luo, X. Liu, and J. Ma, "Network backbone anomaly detection using double random forests based on non-extensive entropy feature extraction," in *Proceedings of the 2013 Ninth International Conference on Natural Computation (ICNC)*, July 2013.

[37] S. Seyedabrishami, A. R. Izadi, H. S. Rayaprolu, and R. Moeckel, "Car Ownership: A Joint Model for Number of Cars and Fuel Types," *Transportation Research Procedia*, vol. 41, pp. 572–576, 2019.

[38] A. Rasaizadi and S. E. Seyedabrishami, "Traffic state prediction by machine learning algorithms for short-term and mid-term prediction time horizons," *Amirkabir Journal of Civil Engineering*, 2021.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9944918, 15 pages
https://doi.org/10.1155/2021/9944918

Research Article

Transferability of a Machine Learning-Based Model of Hourly Traffic Volume Estimation—Florida and New Hampshire Case Study

Przemysław Sekuła,[1,2] **Zachary Vander Laan**,[1] **Kaveh Farokhi Sadabadi**,[1] **Krzysztof Kania**,[2] **and Sara Zahedian**[1]

[1]*Department of Civil and Environmental Engineering, University of Maryland, College Park, MD, USA*
[2]*Faculty of Informatics and Communication, University of Economics in Katowice, Katowice, Poland*

Correspondence should be addressed to Przemysław Sekuła; psekula@umd.edu

Received 31 March 2021; Revised 19 October 2021; Accepted 3 November 2021; Published 26 November 2021

Academic Editor: Chi-Hua Chen

This paper focuses on the problem of model transferability for machine learning models used to estimate hourly traffic volumes. The presented findings enable not only an increase in the accuracy of existing models but also, simultaneously, reduce the cost of data needed for training the models—making statewide traffic volume estimation more economically feasible. Previous research indicates that machine learning volume estimation models that leverage GPS probe data can provide transportation agencies with accurate estimates of hourly traffic volumes—which are fundamental for both operational and planning purposes—and do so with a higher level of accuracy than the prevailing profiling method. However, this approach requires a large dataset for model calibration (i.e., input and continuous count station data), which involves significant monetary investment and data-processing effort. This paper proposes solutions, which allow the model to be prepared using a much smaller dataset, given that a previously collected dataset, which may be gathered in a different place and time period, exists. Based on a broad selection of experiments, the results indicate that the proposed approach is capable of achieving similar model performance while collecting data for a 5 times shorter time period and utilizing 1/4 of the number of continuous count stations. These findings will help reduce the cost of preparing and maintaining the traffic volume models and render the traffic volume estimations more financially appealing.

1. Introduction

This research investigates the transferability of the artificial neural network (ANN) models applied in hourly traffic volume estimation. The introduced methodology explores the feasibility of training an ANN model to estimate the traffic volume by transferring data from another region when sufficient data are unavailable for the target network. This methodology requires ground truth observations to train the models, but previously trained models can be applied at the entire road network, regardless of the existence of the ground truth data. The estimation of traffic flow parameters is one of the fundamental needs in many of the intelligent transportation system applications as they are used for a wide variety of purposes from the planning to the design and operational stages of highway networks [1–3]. Two of the most crucial inputs required by transportation agencies in order to calculate statewide performance measures are traffic volumes and speeds. There are many well-known methods and solutions for obtaining network-wide speed data, which are already implemented for commercial use. However, network-wide traffic volumes are much more challenging to obtain and thus remain a key missing dimension for quantifying traffic conditions, transportation system performance assessment, and cost-effective management of mobility projects and programs. The current estimation methods require continuous count station (CCS) data and knowledge regarding aggregate volume estimates

such as annual average daily traffic (AADT), making them both expensive and inaccurate in locations where CCS data are unavailable. Moreover, the accuracy of these models tends to decrease when the objective is to estimate traffic volumes with higher temporal granularity (e.g., for 15-minute intervals). These difficulties are generated by a lack of network-wide available data reflecting the spatiotemporal traffic volume changes. The I-95 Corridor Coalition founded the "Volume and Turning Movement" project in 2015, seeking to provide representative volume and turning moving data products and assess their accuracy and feasibility. During the first phase of the project, it was shown that an ANN-based volume estimation model leveraging probe data as a key input was able to improve hourly volume prediction accuracy on functional road class (FRC) 1 roads by 26% compared to the currently often-used profiling method [4]. In the second phase, this solution was employed statewide in Florida [5], and extended to the principal, major, and minor arterials. While implementing the model at the statewide level helped us to meet the project's main goals, it required a large calibration dataset to accomplish this. In order to enable this, three months of data from 173 CCSs were collected as the ground truth for training the ANN model to learn the relations between the input variables and actual, observed traffic volumes. However, many states do not have such a robust network of CCSs and thus may not have sufficient calibration data to train a volume estimation model. Nonetheless, some patterns learned from such a large dataset may not be limited to a specific time or location and may be successfully applied elsewhere. For example, the impact of weather or traffic jams on the same types of roads should be similar regardless of the location. A natural way to circumvent the data limitations is to transfer knowledge from areas where enough data are available to areas where the amount of collected data is insufficient to apply to ML techniques. Due to the significant spatiotemporal variation in traffic volumes, previously trained models cannot provide reliable results in a new study area. However, a small dataset from the study area can be combined with large datasets obtained from other regions to train an ANN model with acceptable accuracy. This paper explores whether target geographies with a small calibration network can leverage a larger, previously collected dataset to enhance the volume estimation performance. To illustrate this, we focused on estimating traffic volumes in New Hampshire—a small state with a limited CCS network—by utilizing the previously described Florida dataset, which is four times the size.

The contribution of this article is twofold. We proposed a solution that enables traffic volume estimations even with small datasets, provided that the other available large datasets or previously trained models are leveraged. We also designed, explored, analyzed, and assessed different techniques that allow leveraging available large external datasets or previously trained models to improve traffic volume estimations when only a small dataset for a particular area and time is available. As the cost of the data is the main factor that impedes wide usage of traffic volume estimations, these contributions are of significant practical importance.

The remainder of the paper is organized into five sections. First, the literature review is presented to show how the proposed approach fits into the existing traffic volume estimation research. Afterwards, the data used for the analysis are presented, followed by a description of the model—including hyperparameter selection and details of the training procedure. Subsequently, the experiments and results are discussed and, finally, conclusions are drawn, as several extensions of this work are proposed.

2. Literature Review

The existing literature contains many approaches for estimating traffic volumes [2, 4]. One possible way to categorize these studies is to divide them into two groups of parametric and nonparametric methods [6, 7]. Parametric methods use linear and nonlinear regression or the autoregressive integrated moving average (ARIMA) and their modifications to analyze historical data. The main limitation of parametric methods is that they yield the best results in analyzing stable phenomena with linear relationships between parameters.

As traffic shows features of chaotic systems with nonlinear relationships [8], the researchers' attention turned more to nonparametric methods. They can approach any nonlinear characteristics of the traffic flow data. This group includes many different methods that are mostly concerned with AADT estimations. Reference [9] used support vector machine (SVM) models for AADT prediction, [10] employed artificial neural networks (ANN) and SVM to predict AADT volumes, [11] also employed SVM-based models, [12] and employed classification and regression trees to predict short-term AADT volumes. Decision trees were also used by [13] in intersection traffic prediction. Reference [14] proposed a solution based on genetic algorithms, and a number of publications ([15–18]) used Bayesian networks in predicting traffic flow. Comparative analyses ([19–22]) show that while parametric methods can be used under certain conditions, nonparametric methods are better in others. In addition, there are many hybrid approaches that employ various methods simultaneously, e.g., [23–26].

Among nonparametric solutions, ANNs have been growing in popularity ever since being proposed to estimate motion parameters [27]. The recent advancement in massive data availability has made it possible for ANN models to achieve better results than models based on statistical methods [28]; thus they are widely used to predict traffic flow and AADT. Reference [29] used fuzzy networks to deal with the uncertainty of spatiotemporal data features in traffic flow prediction. Reference [30] handled the problem of missing data in forecasting AADT using recurrent neural networks. Reference [31] examined the possibility of estimating AADT from a one-week dataset using the ANN-fuzzy approach. An extensive literature review related to the use of ANNs for traffic flow prediction was included in [32]. Moreover, multiple recent studies have leveraged ANNs to address the hourly traffic volume estimation. Reference [33] used an ANN to estimate hourly traffic volume using continuous count station features as direct inputs to their model.

Reference [34] estimated freeway off-ramps hourly traffic volume using an ANN. This paper illustrated that the off-ramp volume estimation can achieve acceptable accuracy when proper data are fed into the model. An exhaustive review and classification of the deep learning methods and models used in the estimation of road traffic parameters is also included in [2].

Although ANNs achieve good results, a lot of data are necessary, and meeting this requirement is usually expensive and time-consuming. According to [35], one of the fundamental problems of using predictive models is the limited possibility of using them outside the area or period of study in which it was conducted. A potential solution to the problem of insufficient data is to obtain additional data from other available sources, such as GSM stations [36], GPS data [1], social media, or applications installed voluntarily on mobile devices [37]. Although Big Data technologies and open data sources have made the data problem easier to solve than ever before, in many cases it still remains an issue [38]. The second approach assumes that it is possible to transfer data and models developed in one spatiotemporal context to another location. Transferability can be investigated in the temporal and spatial dimensions. Temporal transferability is the possibility of using observations or models trained in a given time window to estimate parameters in other periods. Spatial transferability means the possibility of using observations or models trained for one area in another location [39]. The spatiotemporal transferability of forecasting models is of considerable practical interest. It can save time and money on gathering data or preparing the model itself and overcoming the lack of a trained personnel [40]. This issue has been of interest in the context of traffic volume estimation for decades (see for instance [3, 40–43]). For machine learning models, this problem is defined as transfer learning. The transfer of data or knowledge from one domain to another is a way to solve the problem of insufficient training data, which is particularly important when using ANNs [44]. Reference [45] demonstrated that the size of the dataset is crucial for the accuracy of traffic forecasts in the ANN-based models and, in terms of results, it is more important than other factors. Hence, the problem of using data or knowledge of traffic patterns acquired in one place at different locations still needs a solution [7, 46]. The current study aims to address this problem for hourly traffic volume estimation. This paper explores the viability of using existing CCS data from Florida to help calibrate an hourly traffic volume estimation model in New Hampshire, where a much smaller calibration dataset is available. In particular, the following two specific issues were explored concerning the model performance:

Using significantly fewer CCSs in the target state (New Hampshire) to calibrate the model

Collecting calibration data from the target state for a shorter time period.

The proposed solution can be classified as transfer learning based on mapping [44], which consists of combining data sources from the source and destination domain, thus creating a new (larger) dataset. This method is based on the assumption that although the data from both domains differ, this new set of data contains sufficient features of the target domain for accurate forecasting. Namely, it applies a strategy named fine-tuning without freezing transferred layers described in [3].

3. Data

Two datasets were used to test the impact of model transferability: Florida and New Hampshire. The Florida dataset encapsulates October–December 2016, while New Hampshire data are from June–September 2017. The Florida dataset is over four times larger than the corresponding New Hampshire dataset (that is, it contains data from four times more Continuous Count Stations), but apart from the location and size, the structure of both datasets is the same. Each dataset is organized spatially by the traffic message channel (TMC) segments and temporally at the hour level. Table 1 summarizes the ground truth values and input variables used for model training and development, which are further described below.

3.1. Ground Truth Data. Hourly traffic volumes from continuous count stations were used as a ground truth (expected output) for the neural network volume estimation models. 173 traffic sensors were used in Florida, while 42 were used in New Hampshire, and these stations were located at all types of major roads (motorways, highways, major and minor arterials). To obtain counts on the TMC network (i.e., the road network used for analysis), each unique count station and traffic direction was mapped to a corresponding TMC segment via GIS analysis.

3.2. Input Features.

Vehicle Probe Counts. The hourly aggregated vehicle probe volumes were obtained based on the raw global positioning system (GPS) data provided by a probe data vendor. The raw waypoints were initially snapped to the XD road segments by the provider. To remain consistent with other data sources, we used a bridge between the XD and TMC segment definitions to match the waypoints with the TMC segments and then aggregated the data at an hourly level. Additionally, each vehicle was associated with one of the three weight classes (class 1: below 14,000 lbs, class 2: between 14,000 and 26,000 lbs, and class 3: above 26,000 lbs), and the probe volumes for each weight class were provided separately to the model. The median of all penetration rates (based on a comparison of the vehicle probe counts with the corresponding continuous count stations) was 2.19% in Florida and 2.3% in New Hampshire.

Vehicle Probe Speeds. The average hourly speed estimates based on GPS data were obtained from RITIS (the regional integrated transportation information

Table 1: Ground truth and input variables used for model development.

Name	Category	Data type	Description
Ground truth volumes			
CCS traffic counts		Integer (≥0)	Hourly traffic counts from continuous count stations
Input data features			
Probe counts (weight class 1)	Probe counts	Integer (≥0)	Number of unique probe counts from weight class 1 vehicles (<14k lbs)
Probe counts (weight class 2)		Integer (≥0)	Number of unique probe counts from weight class 2 vehicles (14k–26k lbs)
Probe counts (weight class 3)		Integer (≥0)	Number of unique probe counts from weight class 3 vehicles (>26k lbs)
Probe count null flag		Binary (0, 1)	Indicator variable used to identify persistent missing probe data
Avg probe speed	Probe speeds	Float (≥ 0)	Average (harmonic) probe speed over 1 hour time period (mph)
Reference speed		Float (≥ 0)	Estimated freeflow speed for segment (mph)
Avg travel time		Float (>0)	Avg time to travel across segment (seconds)
Probe speed null flag		Binary (0, 1)	Indicator variable used to identify persistent missing probe speed/travel times
Temperature	Weather	Float (≥ 0)	Temperature in degrees Fahrenheit
Temperature null flag		Binary (0, 1)	Indicator variable used to identify whether temperature is missing
Dewpoint		Float (≥ 0)	Dew point in degrees Fahrenheit
Dewpoint null flag		Binary (0, 1)	Indicator variable used to identify whether dewpoint is missing
Relative humidity		Float (≥ 0)	Relative humidity (%)
Relative humidity null flag		Binary (0, 1)	Indicator variable used to identify whether relative humidity is missing
Visibility		Float (≥ 0)	Visibility (miles)
Visibility null flag		Binary (0, 1)	Indicator variable used to identify whether visibility is missing
Precipitation		Float (≥ 0)	Precipitation (inches)
Precipitation null flag		Binary (0, 1)	Indicator variable used to identify whether precipitation is missing
Segment length	Road characteristics	Float (≥0)	Length of segment in miles
OSM road class		Binary (0, 1)	6 one-hot encoded variables representing possible OpenStreetMap road classes
OSM lanes		Float (≥0)	OSM-based estimate of number of directional lanes on the road
OSM lanes null flag		Binary (0, 1)	Indicator variable used to identify whether OSM lanes is missing
Structure type		Categorical	Indicates presence of a special facility (bridge, tunnel, causeway)
HPMS through lanes		Float (≥0)	HPMS-based estimate of number of total lanes on the road
Functional system (HPMS)		Binary (0, 1)	3 one-hot encoded variables for HPMS functional classes 1, 2, 3+
Functional road class (TMC)		Binary (0, 1)	3 one-hot encoded variables for TMC functional classes 1, 2, 3+
AADT		Float (≥0)	HPMS annual avg daily traffic associated with TMC segment
AADT (single)		Float (≥0)	Single-unit truck and bus AADT associated with TMC segment
AADT (combination)		Float (≥0)	Combination truck AADT associated with TMC segment
Hour	Temporal	Binary (0, 1)	24 one-hot encoded variables representing hours 0–23
Weekend		Binary (0, 1)	2 one-hot encoded variables for identifying Saturday and Sunday
Volume profile estimate	Other	Float (≥0)	Time-of-day volume estimate based on TTI profiling method
Volume profile null flag		Binary (0, 1)	Indicator variable used to identify whether volume profile estimate is missing

system) [47]. RITIS was created and is maintained by the Center for Advanced Transportation Technology at the University of Maryland, College Park and provides visual analytics and data query capabilities for industry-sourced probe data.

Weather Data. Weather features were extracted from all permanent weather stations using data archived by the Iowa Environmental Mesonet [48] and assigned to each TMC segment based on spatial proximity. Initial tests suggested that the most important weather features are precipitation, temperature, visibility, and humidity.

Road Characteristics. Infrastructural characteristics were extracted from both the National Performance Measurement Research Dataset (NPMRDS) TMC shapefile and the Open Street Map (OSM) road network. As the OSM maps use a different network topology, the OSM road characteristics (road classification and number of lanes) first had to be conflated to the TMC map, which was conducted using an automated conflation algorithm developed for this purpose. The final road characteristic features used for each TMC segment include information regarding road classification, number of lanes, segment length, and reference speed, as well as historical average annual daily traffic values associated with each TMC segment (obtained from the NPMRDS TMC shapefile).

Temporal Data. Information concerning the hour of the day and the day of the week (working day/Saturday/Sunday) was also considered for each data point in order to account for temporal traffic patterns.

Other. Hourly volume profiles were derived by applying the widely used profiling method [49]. This method transforms AADT estimates derived from the highway performance monitoring system [50] into hourly volume profiles based on historic speeds available in RITIS.

4. Model

In previous research [4], it was shown that a fully connected (dense) neural network with three hidden layers yields the best volume estimation model performance. The structure of this network is presented in Figure 1 and is used for all subsequent experiments.

4.1. Training Procedure. The mean absolute error (MAE) was selected as the loss function (i.e., the function that is minimized by the learning algorithm). Although a less popular choice than the mean squared error (MSE)—which tends to place greater emphasis on higher volumes, MAE was selected because it provides a good trade-off between $MAPE$ and R^2 performance metrics, which were used to assess the tested models and approaches. Additionally, MAE was used in previous research, therefore, the selection of MAE makes it easier to compare the results.

The model was trained with the Adam algorithm [51] proposed by Diederik P. Kingma and Jimmy Ba in 2014. Among many advantages, such as quick convergence, computational efficiency, and intuitive hyperparameters, the Adam algorithm is robust in terms of hyperparameter settings and usually does not require much hyperparameter tuning—a feature that was particularly important due to the number of experiments required. Overall, we trained 546 models, which made it impossible to separately tune the hyperparameters manually for each model. Thus,during initial tests, we discovered that the default hyperparameters ($\alpha = 0.01, \beta_1 = 0.9, \beta_2 = 0.999$) work reasonably well and achieve strong model performance. Additionally, experiments showed that tuning hyperparameters around the default values did not significantly change the results—only the speed of convergence. Furthermore, due to the implementation of Dropout [52] after each hidden layer, the models turned out to be resistant to overfitting. Based on these initial findings, we decided to use the default hyperparameters for all training procedures and train the networks longer than required (i.e., to avoid tuning at the expense of some efficiency). The sample loss plots for the models with the smallest and largest datasets are presented in Figure 2. Both train and validation losses do not significantly decrease in the final few epochs, thus demonstrating that the models were trained long enough, and further training would not have improved the accuracies. On the other hand, the validation losses do not increase with time, which shows that the models do not overfit.

The smallest datasets contained only 1 week of New Hampshire data, while the largest one contained all 3 months of both New Hamshire and Florida data. The charts come from the initial (tuning) experiments, where the New Hampshire data were split into train and validation datasets.

4.2. Partitioning Training and Validation Data. Initial tests were first performed to verify the structure of the network and tune the network hyperparameters. These tests were performed for each experiment with a fixed split of New Hampshire data allocated into the training and validation parts. To avoid data leakage different, continuous count stations were used for training and validation sets. Additionally, the split was made taking into consideration the functional road class (each FRC was represented in the same proportions in the training and validation sets). Next, after determining all the hyperparameters, the full crossvalidation procedure was employed. Each model was trained using data for 41 New Hampshire Continuous Count Stations and tested on the 42nd station. This procedure was repeated 42 times to ensure that all NH data were included in the test dataset. While being time consuming, this approach allowed us to avoid data leakage and to take full advantage of the given datasets.

4.3. Evaluating Model Accuracy. During each iteration of cross-validation, model performance is quantified at the test location via the following error metrics: R^2, MAPE (mean absolute percentage error), and EMFR (Error to Maximum Flow Ratio), a process that is repeated during each experiment. These metrics were also used in previous research [4, 5], which renders further comparisons easier. R^2, MAPE, and EMFR metrics are presented in the following equations, where y_i denotes an actual volume, $\widehat{y}_i$ is a volume estimate, $\overline{y}$ stands for the sample average, $y_{\max}$ represents the maximum observed traffic volume, and n is the number of data points used to compute the metric:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \overline{y}\right)^2}, \tag{1}$$

$$\text{MAPE} = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y}_i - y_i}{y_i}\right|\right) \times 100, \tag{2}$$

$$\text{EMFR} = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y}_i - y_i}{y_{\max}}\right|\right) \times 100. \tag{3}$$

R^2 represents the fraction by which the variance of the errors is less than the variance of the dependent variable. In other words, it shows the percent of variance explained. MAPE expresses accuracy as a relation between an absolute error and the real observed value. MAPE is widely used for traffic volume estimations, mainly due to its ease of interpretation. However, MAPE has a few flaws. Namely, in the case of small volumes, the MAPE can be very high even if the absolute error is relatively small. Due to this, MAPE is highly affected by the time periods when the traffic volumes are small, whilst for planning and operational purposes, the time periods with high traffic volumes are usually much more important. EMFR is used in order to deal with this problem. EMFR is defined as the relation between an absolute error and the maximum observed value. Thus, if the absolute error is much smaller than the maximum observed traffic, the

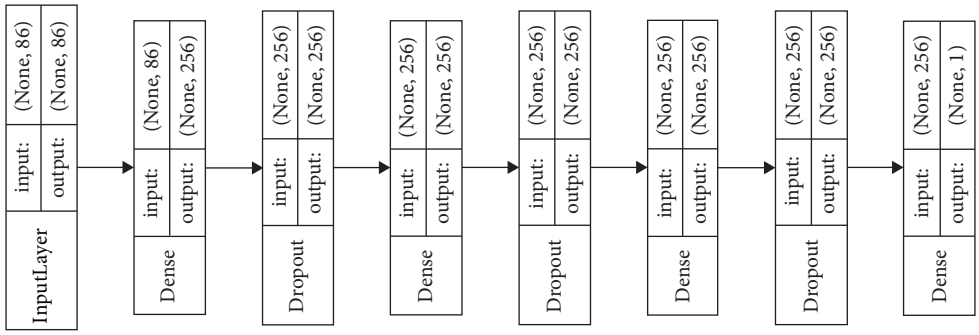


Figure 1: A fully connected (dense) ANN was employed to estimate traffic volumes.

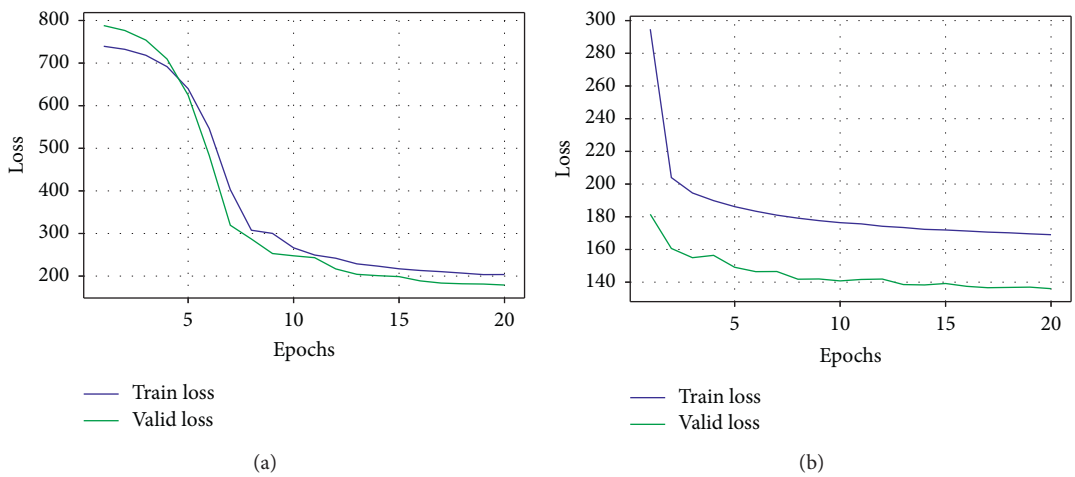


Figure 2: The behavior of the train and validation loss functions for the smallest and largest tested datasets. (a) Smallest dataset and (b) largest dataset.

value of EMFR is also small, regardless of the current traffic. On the other hand, if the observed traffic is close to the maximum, the values of EMFR are close to the values of MAPE. EMFR has also a practical meaning, in the above-mentioned VTM project, this metric was used to define the model quality thresholds (less than 10% EMFR was considered "satisfactory," while less than 5% EMFR was considered "very good").

5. Experimental Results and Discussion

5.1. Experiment 1: Model Comparison. The goal of the first group of experiments was to explore the possibility of using the Florida dataset for New Hampshire predictions. During the experiments, four different approaches were employed and compared. The outline of the experiments is shown in Figure 3.

The "base" model was trained with the New Hampshire data only; no Florida data were used. The detailed results are presented in Table 2 in the *Base* columns. The mean and median R^2s for this approach were 0.72 and 0.82, while the mean and median MAPEs were 43.3% and 26.9%, and the mean and median EMFRs were 8.12% and 7.04%, respectively. The typical (i.e., median) model performance is good

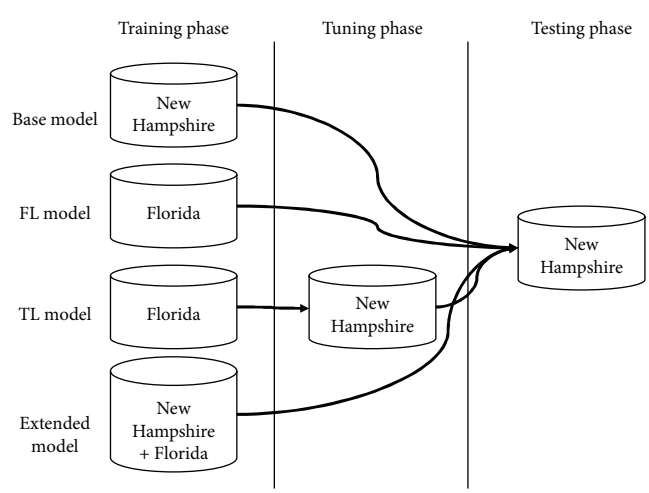


Figure 3: The outline of the experiments—Stage 1.

and is comparable with the results that were obtained for the state of Florida [5] in the previous stage of the project (median R^2: 0.83, median MAPE: 25%). They are also much better than the currently used profiling method (median R^2: 0.60, median MAPE: 55.6%). However, the problem with the

Table 2: The detailed results for different approaches.

	R2				MAPE (%)				EMFR (%)			
	Base	Fl	TL	Ext	Base	Fl	TL	Ext	Base	Fl	TL	Ext
Mean	**0.72**	**0.34**	**0.77**	**0.81**	**43.3**	**61.0**	**34.4**	**34.9**	**8.12**	**12.61**	**7.36**	**7.15**
Std	0.49	1.30	0.30	0.13	47.6	43.8	26.1	29.0	4.03	7.52	2.75	2.23
min	−2.46	−8.31	−1.08	0.06	16.2	27.9	14.1	16.0	4.44	7.52	4.27	4.67
25 per.	0.76	0.54	0.76	0.78	22.7	40.8	20.8	21.4	5.85	8.20	5.93	5.79
50 per.	**0.82**	**0.70**	**0.85**	**0.84**	**26.9**	**49.0**	**26.9**	**25.5**	**7.04**	**9.95**	**6.67**	**6.68**
75 per.	0.88	0.80	0.89	0.88	37.6	59.6	34.0	34.2	8.40	13.96	7.90	7.69
max	0.94	0.92	0.95	0.94	311	286	149	182	28.3	48.0	20.8	20.0

Bold values indicate mean and median values.

"base" model is the outliers, which can be seen in Figure 4—which shows the R^2 metrics with and without outliers—and in Table 2. High standard deviations, a large difference between mean and median R^2s and MAPE s and unacceptable results for the worst continuous count station R^2: −2.74 and MAPE: 326%) emphasize the problem.

Base stands for the model trained on New Hampshire data only, Fl is the model trained on Florida data and tested with NH data, TL stands for transfer learning (the model was trained on the Florida dataset, and then fine-tuned with NH dataset), and *Ext* stands for extended dataset (both Florida and New Hampshire datasets were included into the training set).

Base stands for the model trained on New Hampshire data only, *Florida* is the model trained on Florida data and tested with NH data, TL stands for transfer learning (the model was trained on the Florida dataset, and then fine-tuned with the NH dataset), and *Extended* stands for the extended dataset (both Florida and New Hampshire datasets were included in the training set).

Time series charts (a, b, and c) present the real (blue) and estimated (red) traffic volumes. For an ideal model, these two lines should overlap. Hexabin charts (d, e, and f) present the relations between the real and the estimate traffic volumes. The color of hexagons corresponds to the number of points within each hexagon. As long as the estimates are close to the real values, the points are located on the diagonal line.

Additionally, we selected three locations with EMFR equal to 5%, 10%, and 25% and plotted both time series and hexabin charts for each location (Figure 5). The main purpose of this figure is to facilitate the interpretation of the presented metrics.

The second approach involved training the model with only Florida data, and then testing on the New Hampshire dataset. The detailed results are shown in Table 2 in the *Fl* columns. Although the results are still better than for the profiling method, they are the worst from all the tested models, in terms of both mean and median R^2 s, MAPE s, and EMFR s. This suggests that it is difficult to simply transfer the model from one area (state) and time period, use it in another state and time period, and maintain a high level of estimation accuracy.

The third approach was based on transfer learning and the fine tuning procedure. It was similar to the second one, but this time the model trained with Florida data was fine-tuned with New Hampshire data. The results for this scenario are presented in both Figure 4 and Table 2 (column TL). During the training process, consecutively one, two, or three (all) hidden layers of the model were unfrozen, i.e., the weights of these layers were changed in the fine-tuning procedure. Due to the fact that the model did not overfit, the best results were obtained for all three unfrozen layers. The results presented in the papers are from the network with all the layers unfrozen. The transfer learning based models behave better than the "base" New Hampshire model, for all the metrics, except the median MAPE s, which are equal for both approaches. The transfer learning approach results also in a significant reduction in outliers, which may be noticed in both Figure 4 and Table 2.

Finally, we added the Florida dataset to the training data. For the crossvalidation procedure, each training set for this scenario contained all Florida data, and data for 41 of 42 New Hampshire Continuous Count Stations. The detailed results are shown in Table 2 in the columns *Extended* and in Figure 4. The results indicate that not only is the typical model performance superior to the other approaches (in terms of mean and median R^2 s and MAPE s this approach is the best for three out of four metrics) but it also best deals with outliers.

5.2. Experiment 2: Impact of Dataset Size. The goal of the second set of experiments was to check if it is possible to use both the transfer learning and the extended dataset approaches, as explained in the previous subsection with a dataset, which covers a relatively small time period. These experiments have important practical implementations—the cost of the data depends on the size of the dataset. For these experiments, we compared three approaches—the "base" approach based on the model trained on New Hampshire, the transfer learning approach based on the model pre-trained on the Florida dataset and fine-tuned on the New Hampshire dataset, and the "extended" model—based on using merged Florida and New Hampshire datasets for training.

This time, instead of using all the New Hampshire data in the training procedure, we reduced the time scope of the data; we trained the models using the NH dataset reduced to twelve, eleven, ten, and so on down to one week and (for the extended approach) the entire dataset from Florida. During the training, we repeated the full crossvalidation procedure, and each of the models was tested on a full three months

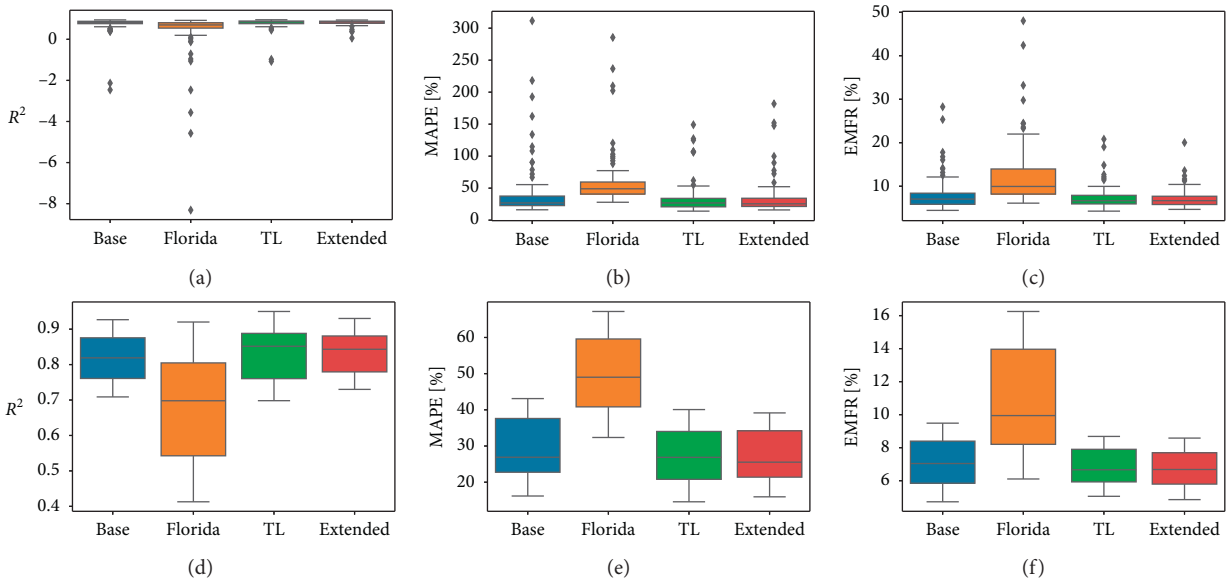


Figure 4: R^2, MAPE, and EMFR metrics for different approaches with and without outliers. (a) R^2 with outliers, (b) MAPE with outliers, (c) EMFR with outliers, (d) R^2 without outliers, (e) MAPE without outliers, and (f) EMFR without outliers.

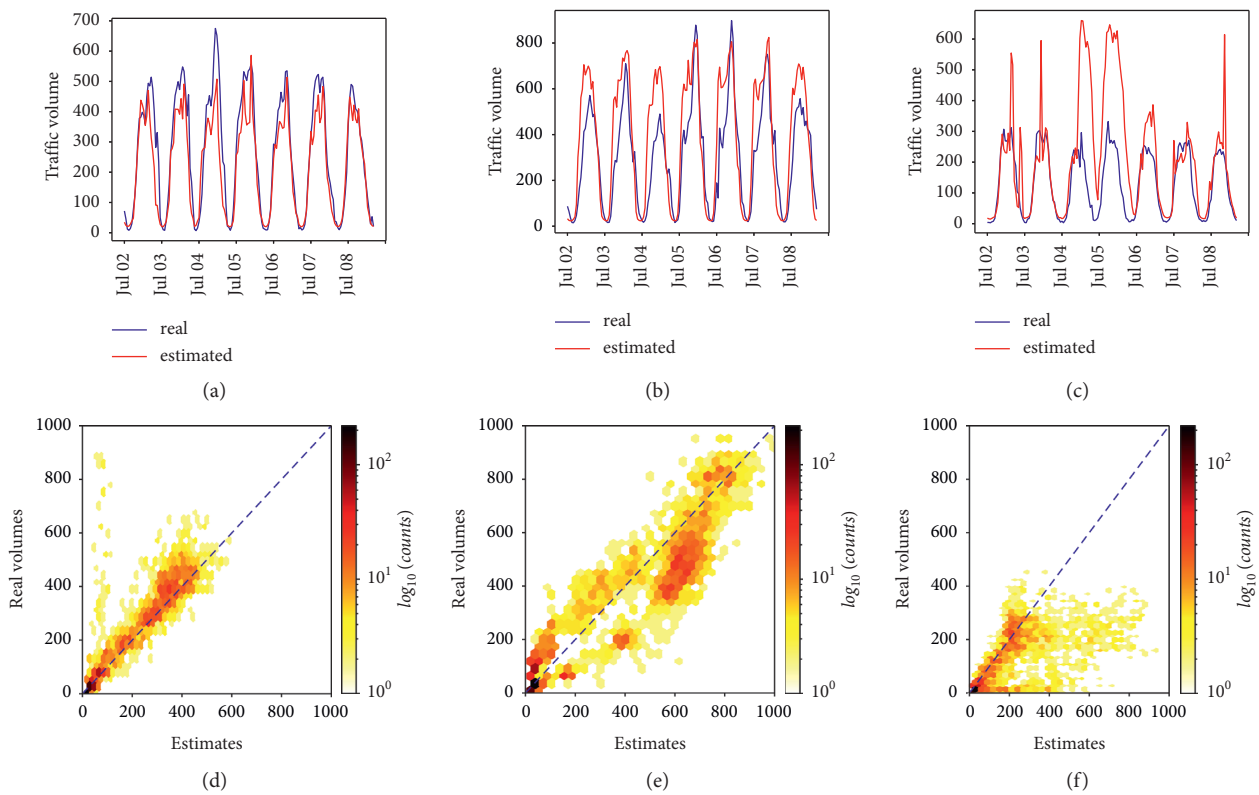


Figure 5: Examples of time series and hexabin plots for different EMFR values. (a) EMFR = 5%, (b) EMFR = 10%, (c) EMFR = 25%, (d) EMFR = 5%, (e) EMFR = 10%, and (f) EMFR = 25%.

New Hampshire dataset that corresponded to the tested continuous count station. The detailed results are presented in Table 3, while a comparison of the models' behavior is illustrated in Figure 6.

The size of the ribbons is proportional to the standard deviation (0.2 ∗ std). Due to the large values of the standard deviation in case of the "base" model, plotting the entire standard deviation makes the figures difficult to read.

Table 3: The detailed results regarding the size of the New Hampshire dataset.

	Base model											
	R2				MAPE (%)				EMFR (%)			
	1 w	5 w	9 w	3 m	1 w	5 w	9 w	3 m	1 w	5 w	9 w	3 m
Mean	**0.08**	**0.63**	**0.69**	**0.72**	**61.9**	**51.9**	**46.9**	**43.3**	**11.46**	**8.75**	**8.27**	**8.12**
Std	3.65	1.00	0.69	0.49	100.5	66.7	55.3	47.6	9.13	5.05	4.46	4.03
min	−23.9	−5.98	−3.80	−2.46	23.4	15.3	16.6	16.2	6.27	4.92	4.62	4.03
25 per	0.53	0.71	0.75	0.76	33.9	24.6	24.1	22.7	8.08	6.25	6.06	5.85
50 per	**0.71**	**0.81**	**0.83**	**0.82**	**43.3**	**31.0**	**27.6**	**26.9**	**9.44**	**7.60**	**7.13**	**7.04**
75 per	0.80	0.87	0.87	0.88	51.1	40.6	39.0	37.6	11.50	8.80	8.17	8.40
max	0.90	0.93	0.95	0.94	798	424	375	311	68.8	38.6	32.7	28.3
	Transfer learning											
	R2				MAPE (%)				EMFR (%)			
	1 w	5 w	9 w	3 m	1 w	5 w	9 w	3 m	1 w	5 w	9 w	3 m
Mean	**0.76**	**0.78**	**0.78**	**0.77**	**35.5**	**36.1**	**37.3**	**34.4**	**7.88**	**7.48**	**7.39**	**7.36**
Std	0.19	0.23	0.27	0.30	24.4	28.8	34.7	26.1	2.44	2.62	2.71	2.75
min	−0.25	−0.53	−0.81	−1.08	15.1	15.6	15.3	14.1	5.01	4.51	5.34	4.27
25 per	0.72	0.76	0.75	0.76	23.5	21.4	20.7	20.8	6.29	5.84	5.78	5.93
50 per	**0.81**	**0.84**	**0.85**	**0.85**	**30.2**	**28.5**	**28.3**	**26.9**	**7.21**	**6.89**	**6.72**	**6.67**
75 per	0.87	0.88	0.89	0.89	36.2	37.7	35.7	34.0	8.37	7.94	8.06	7.90
max	0.94	0.95	0.95	0.95	166	178	191	149	17.3	19.2	20.0	20.8
	Extendedsnmodel											
	R2				MAPE (%)				EMFR (%)			
	1 w	5 w	9 w	3 m	1 w	5 w	9 w	3 m	1 w	5 w	9 w	3 m
Mean	**0.77**	**0.80**	**0.80**	**0.81**	**36.5**	**36.0**	**32.9**	**34.9**	**7.83**	**7.28**	**7.34**	**7.15**
Std	0.13	0.11	0.13	0.13	23.6	28.4	19.9	29.0	1.72	1.67	2.28	2.23
Min	0.10	0.32	0.23	0.06	17.3	16.2	16.0	16.0	4.91	5.03	4.70	4.67
25 per	0.74	0.77	0.75	0.78	23.4	22.6	22.0	21.4	6.55	6.06	5.85	5.79
50 per	**0.80**	**0.82**	**0.84**	**0.84**	**29.7**	**26.5**	**28.0**	**25.5**	**7.76**	**6.98**	**6.64**	**6.68**
75 per	0.86	0.81	0.88	0.88	37.0	34.3	36.1	34.2	8.77	7.93	7.77	7.69
max	0.92	0.94	0.94	0.94	146	190	138	182	12.5	12.3	17.2	20.0

Bold values indicate mean and median values.

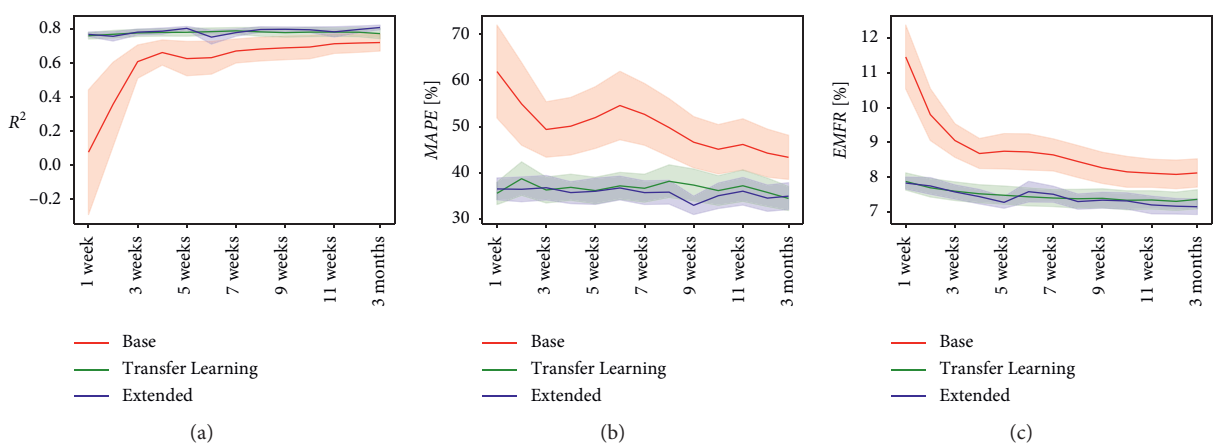


Figure 6: Mean (a) R^2, (b) MAPE, and (c) EMFR metrics depending on the size of the dataset.

The accuracy of the "base" model depends primarily on the size of the New Hampshire training dataset. All the metrics show that the performance of the model diminishes when the size of the dataset is reduced. On the other hand, the accuracy of the "transfer learning" and "extended" models do not significantly change with a change in the size of The New Hampshire training dataset. It shows that by leveraging a larger dataset collected elsewhere, it is possible to achieve good results with as little as one week of data. Moreover, the results for both the "transfer learning" and the "extended" approaches with 1 week of New Hampshire training data are better than the results of the "base" model trained on three months of data, thus emphasizing on the practical usefulness of the proposed solutions.

The "Extended" approach turned out to be slightly more accurate than the one based on transfer learning. However, the differences are not that significant, and the transfer learning-based approach has two advantages. First, due to

Table 4: The detailed results for different models.

	Mean			Median (%)		
	EMFR (%)	MAPE (%)	R^2	EMFR	MAPE (%)	R^2
Dense	6.29	23.6	0.882	6.22	20.8	0.882
RF	6.77	28.7	0.839	6.39	26.4	0.875
LR	11.61	63.8	0.556	8.77	52.9	0.736
BR	9.85	60.9	0.708	8.77	50.4	0.779
PR	6.9e7	7.05e8	−9.3e12	7.43	36.6	0.841

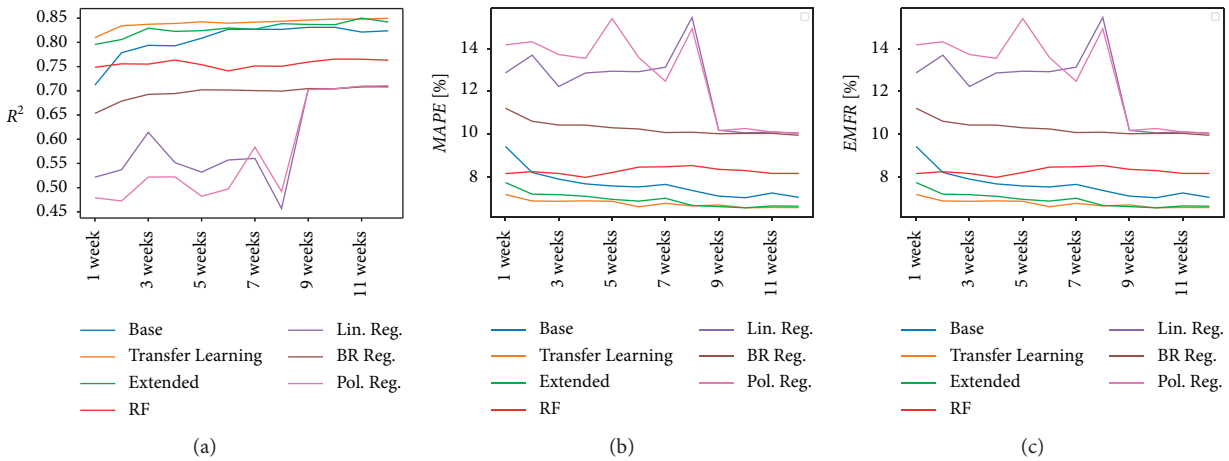


Figure 7: Median (a) R^2, (b) MAPE, and (c) EMFR metrics depending on the size of the dataset.

Table 5: The detailed results for different models—one week long training dataset.

	Mean			Median (%)		
	EMFR (%)	MAPE (%)	R^2	EMFR	MAPE (%)	R^2
Base	11.46	61.9	0.078	9.44	43.3	0.712
TL	7.88	35.1	0.761	7.21	30.2	0.810
Extended	7.84	36.4	0.769	7.76	29.7	0.796
RF	9.85	42.9	0.627	8.18	34.0	0.749
LR	4.1e12	4.4e13	−1.0e24	12.87	75.7	0.522
BR	12.62	71.7	0.429	11.22	56.4	0.653
PR	3.1e12	3.3e13	−5.8e23	14.17	88.0	0.479

the smaller dataset, the training process is much faster for the transfer learning-based approach than for the "extended" one. Additionally, it is possible to employ transfer learning even without direct access to a larger dataset. For transfer learning, only the previously trained model is necessary. This is important due to data licensing. Some vendors offer data for "limited usage" only. It means that the data purchased by a larger state cannot be directly employed for training the model in other states. Such a situation makes using the "extended" approach impossible, while using the pretrained model and transfer learning approach is still doable.

5.3. Discussion. The first set of experiments suggest that utilizing information from a previous dataset can improve traffic volume estimates in a target location—even if the locations and time periods differ. However, it is very difficult to maintain a model's accuracy when applied directly to a different location and time period than it was trained. Directly applying a model trained in one location (e.g., Florida) to a new target location (e.g., New Hampshire) yields reasonable "typical" performance—implying that, in general, the model captures the relation between input variables and traffic volumes, but suffers from severe outliers. Instead, a more promising approach appears to be fine-tuning the model originally trained in one location using data from the target location—an approach that incorporates patterns in the target location, and thus helps to avoid many outliers. Finally, if available, incorporating all data for training purposes (e.g., adding all Florida data for training a model to estimate New Hampshire volumes) yields the best results.

Given the findings from the first set of experiments—namely, that a previous dataset can be used to improve estimates in a new location, the second set of experiments seek to understand how much data are needed at the target location to achieve acceptable accuracy. Focusing,

in particular, on the "transfer learning" with fine-tuning and "extended" approaches, it finds that given a sufficiently large alternate dataset to leverage (i.e., Florida), both approaches can achieve reasonable results with as little as one week of data in the target (i.e., New Hampshire) location.

Overall, it appears that the machine learning-based method of volume estimation requires large datasets to learn complex spatial and temporal patterns between the input variables and traffic volumes. Given financial limitations and the fact that some states and jurisdictions are small and have limited count stations on various road classes, it can be challenging to collect sufficient data to train a reliable estimation model. However, the approaches highlighted in this paper show that input features and corresponding traffic counts can be leveraged from other locations, and used in conjunction with small amounts of data from the target location to develop better overall models. The results indicate that if possible, the best approach is the "extended" one, which uses all previous data in the training process, and does the best job of eliminating outliers. However, in cases where this is not possible (perhaps due to licensing of previous data), the "transfer learning" with fine-tuning approach can be used. This approach only requires access to a previously trained model (not the raw data), whose hyperparameters are subsequently tuned while training the model based on data from the target location.

6. Conclusion

This paper explores the transferability of volume estimation machine learning models, seeking to understand the extent to which a model trained to predict traffic volumes in one location can be used directly or modified slightly to do so elsewhere. The implications of this research question are significant, as it is expensive to acquire GPS trajectory data—a key model input— and time consuming to preprocess data sources. If existing large datasets can be utilized for training purposes, smaller, less expensive datasets can be purchased in new locations and used to build accurate models—potentially saving transportation agencies time and money.

The experimental results suggest that a key component of model performance is having sufficient data for training. If access to a larger, existing dataset is available, the optimal approach is to train the volume estimation model on all available data—using both locations in the target geography and previously collected data. However, even if the raw dataset is not available from a previous location (perhaps due to licensing restrictions), an existing model can be trained and fine-tuned using a small amount of data from the target locations. Interestingly, the larger dataset used to improve performance appears to be useful even if it comes from different places and time periods, and when the target dataset is as small as one week.

Note that the results provided in this study are limited to the hourly volume estimation model's transferability at the state level. The states whose data were available to test the proposed approach encompass both urban and rural areas with different land-use characteristics. Additionally, all roads used for the analyses are FRC 1 or 2 due to data accessibility limitations. Therefore, overgeneralizing the results to various geographical and road levels should be avoided. However, given the availability of the required data for different locations, it is possible to test the proposed approach in other areas to investigate the extent to which it can be generalized.

A key future direction for research includes investigating how much data need to be collected from continuous count stations in order to constitute a satisfactorily large dataset for transfer learning, and whether there are certain temporal or spatial characteristics that are necessary for the data to be transferable. Additionally, it would be beneficial to understand how often the models should be fine-tuned or retrained in order to optimize performance. Based on the promising results presented in this paper, these research questions will be explored in more detail in future modeling efforts.

Appendix

A. Model Selection

This appendix describes the experiments performed to verify whether the solution selected in [4] (a fully connected artificial neural network) is still the most suitable for the presented task. Similar to Section 5, two experiments were carried out.

First experiment—carried out on the entire dataset. The primary purpose of this experiment was to analyze alternative models and compare them with artificial neural networks. This experiment corresponds to the experiment described in Section 5.1. *Experiment 1: model comparison.*

Second experiment—carried out on reduced datasets. It was conducted to check if the conclusions based on the results from the first experiment are valid for much smaller datasets. This experiment corresponds to the experiment described in Section 5.2. *Experiment 2: impact of the dataset size.*

A.1. Analyzed Models. In the first step of the presented analysis, the authors analyzed different models and selected some of them for further tests. The following models were selected for future analysis:

Fully connected neural network (dense): The model used in [4]. This model is a neural network built of three fully connected (dense) hidden layers, 256 neurons in each layer. The hidden layer neurons use ELU activation function.

Random forest (RF): In our experiments, this model performs only slightly worse than the dense neural network. RF models tend to overfit, so we applied regularization; namely, we limited the minimum samples per leaf (values 5, 10, 15, 95, and 100 were tested). The model with the smallest median EMFR was selected. This model was employed for both experiments.

Linear regression (LR): A basic linear regression model was used as a baseline for our experiments.

Bayesian ridge regression (BR): an approach to linear regression that tries to solve the problem of poorly distributed data by using probability distributions rather than point estimates.

Polynomial regression (PR): A variation of linear regression that uses polynomial features. Technically, the model is linear, but additional features are provided to ensure the polynomial output. For example, if the original input features were X_1 and X_2, and we are interested in the second degree polynomial, the following set of features shall be provided as a model input: $X_1, X_2, X_1 * X_2, X_1^2, X_2^2$.

The following models were not considered in this comparison:

Support vector machines (SVM): SVM is a very powerful solution, but it does not scale well to large datasets. Our training dataset contained over 830,000 samples, thus it was impossible to train an SVM model with a radial basis function (RBF) kernel that ensures the best performance of this approach. In fact, it turned out that with the Sci-kit Learn library, it was also impossible to train an SVM model using polynomial or linear kernels that are expected to generate much worse results than an RBF kernel. Therefore, we did not include this class of models in our analysis.

Long-short-term memory neural networks (LSTM)—LSTM networks seem to be the most suitable architecture for the presented task, as they can generate estimates using also the previously observed features as inputs, and simultaneously, not excessively increase the complexity of the networks. During previous (unpublished) research, we used LSTM networks and achieved slightly better results than with dense neural networks. However, there are two main reasons that prevented us from using these models in the presented research.

Depending on the size of the train dataset, it takes 6–48 hours to train the model. Overall, we trained more than 500 models; thus, it would not be possible to repeat the entire procedure with LSTM models in a reasonable timespan.

Although LSTM models are slightly better than dense neural networks, we also discovered that for the given type of data they are very susceptible to small changes in the hyperparameters. The gist of our research was to discover how the models behave in given different data scenarios. Had we used an LSTM model and got worse results for some scenario, we could not be sure whether this would be the problem with the scenario or the model hyperparameters. One can argue that we could fine-tune the hyperparameters for each scenario, but given the aforementioned training time, it would take far too long.

A.2. Datasets. All the models were trained and tested with the same datasets. During the first experiment, the data were divided into two subsets:

The train dataset comprising of the entire Florida dataset (172 continuous count stations, 691,486 data points) and a part of the New Hampshire dataset (25 Continuous Count Stations, 140,306 data points).

The test dataset comprising of 17 continuous count stations (35,098 data points) located in New Hampshire.

Both train and test datasets contained the features described in Section 3.2.

In the second experiment, only the New Hampshire dataset (25 continuous count stations, 175,404 data points) was selected. To perform this experiment, the authors followed the procedure described in Section 5.2. *Experiment 2: impact of dataset size*, namely, the size of the dataset was gradually reduced from twelve weeks to one week. Each time, the entire crossvalidation procedure was carried out to determine the models' accuracy.

A.3 Results. The final results of the first experiment are presented in Table 4. For random forest, the authors trained the models for different minimum samples per leaf values and chose the best model (the model with the smallest median EMFR value).

A fully connected (dense) neural network turned out to be the best approach, although the results generated by the random forest model were very alike. This is consistent with the results presented in [4]. The results generated by the polynomial regression model are surprising. The medians of the metrics are not unexpected, although the means are very poor. Overall, the model behaved well, but it completely misestimated the traffic for one CCS, in one direction. The authors believe that this behavior is an overfitting problem, which was caused by fitting the model with too many features. To generate a second degree polynomial model as presented in the table, the authors used 3,240 features instead of the 79 used in all other models. Due to the observed overfitting, the authors did not test higher degree polynomial models (for example, to generate a third degree polynomial model, 88,560 features are needed, and a fourth degree polynomial model requires 1,837,620 features).

The second experiment's results overview is presented in Figure 7. Additionally, Table 5 consists of the detailed results for the smallest (one week long) dataset. First of all, both linear and polynomial regression could not deal with outliers. Similar to polynomial regression in the first experiment, they completely misestimated the traffic for one CCS in one direction, which resulted in terrible mean results. The median results presented in Figure 7 show that these models cannot handle the datasets smaller than 9 weeks of data. Bayesian ridge regression results were stable and did not deteriorate heavily with the train dataset size reduction. Moreover, Bayesian Ridge Regression managed to deal with

outliers, regardless of the size of the dataset. However, these results were significantly worse than the ANN-based results. Random forest turned out to be the best of the alternative models. For a two-week-long training set, RF results were similar to the results obtained with a "base" artificial neural network, and for a one-week-long dataset, they were even better. However, regardless of the size of the dataset, RF results were worse than the results of both approaches that leveraged the Florida dataset, namely, extended and transfer learning-based models.

The first three rows of the table present results obtained with dense neural networks. *Base* stands for the model trained on New Hampshire data only, TL stands for transfer learning (the model was trained on the Florida dataset, and then fine-tuned with NH dataset), and *Extended* stands for the extended dataset (both Florida and New Hampshire datasets were included into training set).

Data Availability

The data used to support the findings of this study are not available due to third-party rights.

Disclosure

An early version of this manuscript has been published at the 99th Annual Meetingof the Transportation Research Board. An earlier version of this paper was presented at the TRB annual meeting.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Acknowledgments

The authors would like to thank the I-95 Corridor Coalition for funding this work through the Volume and Turning Movement Project. The authors would also like to thank the Florida DOT and New Hampshire DOT for providing their CCS data. This support is gratefully acknowledged, but it implies no endorsement of the findings. This work was supported by the I-95 Corridor Coalition through the Volume and Turning Movement (VTM) project.

References

[1] H.-h. Chang and S.-h. Cheon, "The potential use of big vehicle GPS data for estimations of annual average daily traffic for unmeasured road segments," *Transportation*, vol. 46, no. 3, pp. 1011–1032, 2019, https://link.springer.com/10.1007/s11116-018-9903-6.

[2] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: methods, analysis and future directions," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2021.

[3] J. Li, F. Guo, A. Sivakumar, Y. Dong, and R. Krishnan, "Transferability improvement in short-term traffic prediction using stacked LSTM network," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102977, 2021.

[4] P. Sekuła, N. Marković, Z. Vander Laan, and K. Farokhi Sadabadi, "Estimating historical hourly traffic volumes via machine learning and vehicle probe data: a Maryland case study," 2017.

[5] N. R. E. L. CATT, "Volume & turning mvmt project," 2018, https://i95coalition.org/wp-content/uploads/2015/02/I-95CC-VTM-SCMtg-02-13-2018-final-combined.pdf?x70560.

[6] N. Slimani, I. Slimani, N. Sbiti, and M. Amghar, "Traffic forecasting in Morocco using artificial neural networks," *Procedia Computer Science*, vol. 151, pp. 471–476, 2019, https://www.sciencedirect.com/science/article/pii/S1877050919305265.

[7] S. Deng, S. Jia, and J. Chen, "Exploring spatial-temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data," *Applied Soft Computing*, vol. 78, pp. 712–721, 2019, https://www.sciencedirect.com/science/article/pii/S1568494618306082.

[8] P. Shang, X. Li, and S. Kamae, "Chaotic analysis of traffic time series," *Chaos, Solitons & Fractals*, vol. 25, no. 1, pp. 121–128, 2005, https://www.sciencedirect.com/science/article/pii/S0960077904006472.

[9] M. Castro-Neto, Y. Jeong, M. K. Jeong, and L. D. Han, "AADT prediction using support vector regression with data-dependent parameters," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2979–2986, 2009.

[10] S. Islam, "Estimation of annual average daily traffic (AADT) and missing hourly volume using artificial intelligence," Ph.D. thesis, Clemson University, Clemson, SC, USA, 2016.

[11] Y. Zhang and Y. Liu, "Traffic forecasting using least squares support vector machines," *Transportmetrica*, vol. 5, no. 3, pp. 193–213, 2009 b.

[12] Y. Xu, Q.-J. Kong, and Y. Liu, "Short-term traffic volume prediction using classification and regression trees," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 493–498, 2013.

[13] W. Alajali, W. Zhou, S. Wen, and Y. Wang, "Intersection traffic prediction using decision tree models," *Symmetry*, vol. 10, no. 9, p. 386, 2018.

[14] B. Abdulhai, H. Porwal, and W. Recker, "Short-term traffic flow prediction using neuro-genetic algorithms," *Intelligent Transportation Systems Journal*, vol. 7, no. 1, pp. 3–41, 2002.

[15] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[16] S. Yang and G. A. Davis, "Bayesian estimation of classified mean daily traffic," *Transportation Research Part A: Policy and Practice*, vol. 36, no. 4, pp. 365–382, 2002, https://www.sciencedirect.com/science/article/pii/S0965856401000088.

[17] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using bayesian networks," *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 482–509, 2008, https://www.sciencedirect.com/science/article/pii/S0191261507001300.

[18] A. Pascale and M. Nicoli, "Adaptive bayesian network for traffic flow prediction," in *Proceedings of the Statistical Signal Processing Workshop*, pp. 177–180, Nice, France, June 2011.

[19] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002, https://www.sciencedirect.com/science/article/pii/S0968090X02000098.

[20] K. Jha, N. Sinha, S. S. Arkatkar, and A. K. Sarkar, "A comparative study on application of time series analysis for traffic

forecasting in India: prospects and limitations," *Current Science*, vol. 110, no. 3, p. 373, 2016.

[21] Y. Zhang and Y. Liu, "Comparison of parametric and non-parametric techniques for non-peak traffic forecasting," *International Journal of Mathematics and Computer Science*, vol. 3, no. 3, pp. 172–178, 2009 a, https://publications.waset.org/vol/27.

[22] J. Salotti, S. Fenet, R. Billot, N.-E. El Faouzi, and C. Solnon, "Comparison of traffic forecasting methods in urban and suburban context," in *Proceedings of the 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 846–853, Volos, Greece, November 2018.

[23] X. Luo, L. Niu, and S. Zhang, "An algorithm for traffic flow prediction based on improved sarima and ga," *KSCE Journal of Civil Engineering*, vol. 22, no. 10, pp. 4107–4115, 2018.

[24] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.

[25] L. Li, B. Ran, J. Zhu, and B. Du, "Coupled application of deep learning model and quantile regression for travel time and its interval estimation using data in different dimensions," *Applied Soft Computing*, vol. 93, Article ID 106387, 2020.

[26] L. Li, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Traffic speed prediction for intelligent transportation system based on a deep feature fusion model," *Journal of Intelligent Transportation Systems*, vol. 23, no. 6, pp. 605–616, 2019, https://www.tandfonline.com/doi/full/10.1080/15472450.2019.1583965.

[27] S. C. Sharma, P. Lingras, F. Xu, and G. X. Liu, "Neural networks as alternative to traditional factor approach of annual average daily traffic estimation from traffic counts," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1660, no. 1, pp. 24–31, 1999.

[28] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[29] W. Chen, J. An, L. Renfa et al., "A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features," *Future Generation Computer Systems*, vol. 89, 2018.

[30] Z. Khan, S. M. Khan, K. Dey, and M. Chowdhury, "Development and evaluation of recurrent neural network-based models for hourly traffic volume and annual average daily traffic prediction," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 7, pp. 489–503, 2019.

[31] M. Gastaldi, G. Gecchele, and R. Rossi, "Estimation of annual average daily traffic from one-week traffic counts. a combined ann-fuzzy approach," *Transportation Research Part C: Emerging Technologies*, vol. 47, no. 1, pp. 86–99, 2014.

[32] P. Kanestørm, "Traffic flow forecasting with deep learning. m.s. thesis ntnu, 2017," 2017, https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2445851/14739_FULLTEXT.pdf Master thesis.

[33] S. Zahedian, P. Sekuła, A. Nohekhan, and Z. Vander Laan, "Estimating hourly traffic volumes using artificial neural network with additional inputs from automatic traffic recorders," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 3, pp. 272–282, 2020.

[34] A. Nohekhan, S. Zahedian, and A. Haghani, "A deep learning model for off-ramp hourly traffic volume estimation," *Transportation Research Record*, vol. 2675, no. 7, 2021.

[35] X. Cao, X. Zhu, Z. Tian, J. Chen, D. Wu, and W. Du, "A knowledge-transfer-based learning framework for airspace operation complexity evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 61–81, 2018, https://www.sciencedirect.com/science/article/pii/S0968090X18309860.

[36] H. Dong, J. Man, L. Jia, X. Wang, Y. Qin, and K. Liu, "Traffic speed estimation using mobile phone location data based on longest common subsequence," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems, ITSC*, pp. 2819–2824, Maui, HI, USA, November 2018.

[37] Z. Zhang, M. Li, X. Lin, and Y. Wang, "Network-wide traffic flow estimation with insufficient volume detection and crowdsourcing data," *Transportation Research Part C: Emerging Technologies*, vol. 121, p. 102870, 2020.

[38] Y. Zhao, H. Zhang, L. An, and Q. Liu, "Improving the approaches of traffic demand forecasting in the big data era," *Cities*, vol. 82, pp. 19–26, 2018, https://www.sciencedirect.com/science/article/pii/S0264275117315081.

[39] K. Andrade, S. Kagaya, K. Uchida, A. Dantas, and A. Nicholson, "Investigating the temporal transferability of transport modal choice models: an approach based on gis data base," *Proceedings of the Eastern Asia Society for Transportation Studies*, vol. 7, p. 80, 2007.

[40] S. Sikder, A. R. Pinjari, S. Srinivasan, and R. Nowrouzian, "Spatial transferability of travel forecasting models: a review and synthesis," *International Journal of Advances in Engineering Sciences and Applied Mathematics*, vol. 5, no. 2-3, pp. 104–128, 2013.

[41] H. Gunn, "Spatial and temporal transferability of relationships between travel demand, trip cost and travel time," *Transportation Research Part E: Logistics and Transportation Review*, vol. 37, no. 2-3, pp. 163–189, 2001.

[42] J. Fox and S. Hess, "Review of evidence for temporal transferability of mode-destination models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2175, no. 1, pp. 74–83, 2010, https://journals.sagepub.com/doi/10.3141/2175-09.

[43] J. Fox, A. Daly, S. Hess, and E. Miller, "Temporal transferability of models of mode-destination choice for the Greater Toronto and Hamilton Area," *Journal of Transport and Land Use*, vol. 7, no. 2, pp. 41–62, 2014.

[44] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," 2018, https://arxiv.org/abs/1808.01974.

[45] N. T. Ratrout and U. Gazder, "Factors affecting performance of parametric and non-parametric models for daily traffic forecasting," *Procedia Computer Science*, vol. 32, pp. 285–292, 2014, https://www.sciencedirect.com/science/article/pii/S1877050914006267.

[46] T. Ma, Z. Zhou, and C. Antoniou, "Dynamic factor model for network traffic state forecast," *Transportation Research Part B: Methodological*, vol. 118, no. C, pp. 281–317, 2018, https://ideas.repec.org/a/eee/transb/v118y2018icp281-317.html.

[47] RITIS, "The regional integrated transportation information system (RITIS)," https://www.cattlab.umd.edu/?portfolio=ritis, 2017.

[48] Iowa State University, 2018, https://mesonet.agron.iastate.edu/.

[49] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "Appendix A: methodology for the 2015 Urban Mobility Scorecard," Texas Transportation Institute, Texas A&M University, 2015, https://static.tti.tamu.edu/tti.tamu.edu/documents/mobility-scorecard-2015-appx-a.pdf Technical Report.

[50] HPMS, 2017, https://www.fhwa.dot.gov/policyinformation/hpms.cfm.
[51] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," http://arxiv.org/abs/1412.6980, 2014.
[52] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, https://arxiv.org/abs/1207.0580.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4486149, 16 pages
https://doi.org/10.1155/2021/4486149

Research Article

A Bayesian Neural Network-Based Method to Calibrate Microscopic Traffic Simulators

Qinqin Chen,[1] Anning Ni,[1] Chunqin Zhang,[2] Jinghui Wang,[1] Guangnian Xiao,[3] and Cenxin Yu[1]

[1]*Department of Transportation Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*School of Civil Engineering and Architecture, Zhejiang Sci-Tech University, Hangzhou 310018, Zhejiang, China*
[3]*School of Economics & Management, Shanghai Maritime University, Shanghai 201306, China*

Correspondence should be addressed to Anning Ni; nianning@sjtu.edu.cn

Received 15 April 2021; Revised 31 August 2021; Accepted 23 October 2021; Published 26 November 2021

Academic Editor: Chi-Hua Chen

Calibrating the microsimulation model is essential to enhance its ability to capture reality. The paper proposes a Bayesian neural network (BNN)-based method to calibrate parameters of microscopic traffic simulators, which reduces repeated running of simulations in the calibration and thus significantly improves the calibration efficiency. We use BNN with probability distributions on the weights to quickly predict the simulation results according to the inputs of the parameters to be calibrated. Based on the BNN model with the best performance, heuristic algorithms (HAs) are performed to seek the optimal values of the parameters to be calibrated with the minimum difference between the predicted results of BNN and the field-measured values. Three HAs are considered, including genetic algorithm (GA), evolutionary strategy (ES), and bat algorithm (BA). A TransModeler case of highway links in Shanghai, China, indicates the validity of the proposed calibration method in terms of error and efficiency. The results demonstrate that the BNN model is able to accurately predict the simulation and adequately capture the uncertainty of the simulation. We also find that the BNN-BA model produces the best calibration efficiency, while the BNN-ES model offers the best performance in calibration accuracy.

1. Introduction

In the process of efficiently assessing the influence of traffic management schemes and even emerging technologies on traffic performance, simulations are powerful tools to simulate the proposed scenes without the need for field experiments [1]. Microscopic traffic simulator is one of the tools, and it outperforms mesoscopic and macroscopic simulators in the simulation of specific driving behavior and the visual display of traffic scenes [2]. However, considering that the traffic conditions vary from place to place, the default values of parameters from simulator developers are inconsistent with the specific characteristics of the places of simulation. Using the default parameters can cause the simulation results to be out of the range of the places of simulation and cause large deviations in the results. In other words, because of the location-dependent parameter values, it is essential to determine a set of parameter values before effectively using microscopic traffic simulators. Thus, it is important to calibrate the parameter values, which enables the simulator to be used for traffic analysis under the preferred background.

The calibration needs to search for the set of parameter values, which ensures the minimum difference between the field-measured values and simulation output values. The calibration is still challenging in microscopic traffic simulation containing high uncertainty of the simulation system and a large number of parameters [3]. In addition, the traditional parameter calibration process requires several runs of simulations. Considering that simulation is based on a certain scale of the road network and complex calculations, an inestimable and long period of time is spent on one

simulation. The number of simulations is affected by calibration convergence. Therefore, finding specific parameter values during the calibration process will be time-consuming and computationally expensive, and it is necessary to improve calibration efficiency.

The research on the parameter calibration method can be divided into two categories according to the research methods, which are nonheuristic methods and heuristic optimization methods. Firstly, previous research mainly focused on how to calibrate based on a variety of nonheuristic methods, such as comparison [4], orthogonal experiments [5], indirect calibration method [6], and Bayesian framework-based technique [7]. The calibration problem inherently seeks and optimizes specific parameter values with the minimum difference between field-measured values and simulation output values. It is efficient to treat the calibration as a simulation-based nonlinear optimization problem [8]. Numerous studies have tried to calibrate the parameters from this aspect. A metamodel-based simulation-optimization framework is newly proposed for calibration [9]. With the help of several heuristics, solving the optimization problem becomes easier. The use of HA in calibration has also attracted much attention from researchers. In HA, GA occupies an important position and cannot be ignored. GA has been widely used in the calibration of microscopic traffic simulators (i.e., FRESIM [10], VISSIM [11–13], AIMSUN [14, 15], and PARAMICS [16, 17]) and traffic microsimulation models [18]. For heterogeneous traffic, GA was suitable for obtaining optimized parameter sets [19]. When exploring the impact of different measures of performance on the calibration results, some scholars still preferred GA [20]. The above research shows that the GA-based method performs well in calibration. Similar to GA, ES is based on the same principle of biological evolution, which has also been used in traffic calibration problems. ES was applied to search for two coefficients in the automatic calibration of traffic surveillance cameras [21]. Some scholars used the covariance matrix adaptation-evolution strategy (CMA-ES) to calibrate dynamic traffic assignment models. The parallel feature of CMA-ES made it suitable for the new large-scale field of applications, not limited to real-time deployment [22]. Additionally, simultaneous perturbation stochastic approximation (SPSA) was also implemented in the calibration framework, which turned out to be fast under numerous inputs [23]. In order to calibrate the discrete demand of multimodal microscopic traffic simulation, the weighted SPSA was investigated [24]. In recent years, BA, as an emerging heuristic algorithm, has mainly been used for continuous optimization [25], combinatorial optimization and scheduling [26], engineering problems [27, 28], and parameter estimation [29]. The above literature proves that BA has significant potential to outperform other algorithms in efficiency and robustness. Since there are various types of HA, it is necessary to compare them. For the calibration of microsimulation models, several metaheuristic algorithms were compared with the manual method. The comparison has shown that HA is superior in saving time and getting better-modeled results [30, 31]. Although BA surpasses GA and particle swarm optimization algorithms in terms of efficiency and accuracy [32], few scholars apply it to the parameter calibration of microscopic traffic simulators or include it in comparative research.

The difficulty of the existing ways is that using heuristics in optimization and calibration needs repeated running of the simulator to obtain outputs, when a different combination of parameter values is being tested. Therefore, calibration still inevitably requires long calculation and simulation time. For the calibration of the microsimulation model, the running time of HA was about 100 hours [30]. Moreover, the GA-based methods took 8.4 hours and about 10 hours to calibrate in two different simulation cases [31, 33]. Although some scholars attempt to save time by using parallel computing technology [34], the research on improving calibration speed is still limited. Reducing the number of simulations in calibration is a possible prerequisite for improving calibration efficiency. Considering that the neural network has been used for prediction in the field of transportation [35], the proposed method substitutes running time-consuming simulations with running the neural network model to gain the predicted simulation outputs in the calibration and thereby greatly improves the efficiency of calibration. It is different from the existing ways. Furthermore, the simulator is a complex nonlinear system. The neural network can fit continuous functions of arbitrary complexity with arbitrary accuracy [36], which can perfectly replace the simulator in a data-driven way. Some studies have made some progress in this area, which also demonstrated the feasibility and benefits of the calibration framework combining neural networks and HA. The applicability of neural networks in calibrating traffic simulators has been studied [37]. An artificial neural network (ANN) set mean target headway and mean reaction time as inputs and outputted the queue length to replace PARAMICS simulation processes. According to the relationship between inputs and outputs given by ANN, GA obtained the calibrated parameter values without simulation. And the consistency between the simulation outputs after calibration and the outputs of ANN verified the validity of the method [38]. Based on VISSIM, a parameter calibration method combining ANN and GA was also proposed. For each parameter set, ANN outputted predictive vehicle speed without simulations. GA took about 1 minute to find parameter values using the trained ANN, but the traditional GA-based method took more than 10 hours. The calibration results reduced the average absolute relative error of the speed in two tested highway sections [33]. Similarly, some scholars have also conducted comparative studies on parameter calibration methods that combine a variety of machine learning models and particle swarm optimization algorithms [39].

The above studies are based on deterministic machine learning models. Nevertheless, probabilistic programming outperforms them in portraying the uncertainty of simulation and driving behavior. BNN has weights that obey the probability distribution to obtain rich prediction results with uncertainty. There are relatively limited studies on the use of BNN in calibration. Only some studies have calibrated the

car-following model in simulation logic based on related ideas. According to the Markov chain Monte Carlo simulation and the Bayesian estimation theory, a stochastic calibration method could estimate the parameter distribution of two car-following models. It was better than the deterministic optimization algorithm from the evaluation of the cost function [40]. Based on real data, probabilistic programming and Bayesian machine learning were applied to the calibration of car-following models. The method showed informative validity and true-to-data uncertainty [41].

The objective of this paper is to develop an efficient calibration method for microscopic traffic simulators. Although the combination of ANN and GA is efficient, the combination of BNN and other HAs (i.e., ES and BA) is expected to continue to improve efficiency and accuracy. In this paper, the combination of BNN and HA is investigated. The purpose of BNN is to achieve that there is no need to run simulations during the calibration process after training the BNN. Before training the BNN, the parameters that have a significant impact on the simulation results are selected as the parameters to be calibrated. Afterward, the data set of simulation results is derived from running enough simulations based on random combinations of values for selected parameters, which is used to train, verify, and test the BNN. Then, the trained BNN can predict simulation outputs by inputting the selected parameter values. Additionally, the trained BNN with the best results becomes the background of optimization problems. Several HAs, including GA, ES, and BA, are investigated to seek optimal parameter values with the minimum difference between the predicted output of BNN and the field-measured data. Specifically, the proposed parameter calibration method includes the combination of BNN and GA (BNN-GA), the combination of BNN and ES (BNN-ES), and the combination of BNN and BA (BNN-BA). They are applied and verified through a real case in Shanghai, China. Because the proposed parameter calibration method is independent of the simulation platform and the scale of the studied road network, it is universal for microscopic traffic simulators and is also expected to be used in a variety of traffic situations.

The innovations of this paper are summarized as follows:

(i) The calibration efficiency of the microscopic traffic simulator is further improved under reliable calibration accuracy by using the proposed method. Calibration is challenging and time-consuming due to a large number of parameters and complicated uncertainties. Compared with the traditional method, the proposed calibration method does not need to run simulation during calibration, which greatly saves time. Moreover, derived from the similar efficient calibration framework using neural networks and HA [33, 37–39], the method combining BNN and HA also performs better in efficiency and error.

(ii) BNN is used in the proposed calibration method. Superior to widely used deterministic machine learning models, BNN is rarely used to predict simulation results, but it can capture uncertainty during calibration.

(iii) GA is commonly used for calibration, but efficient ES and BA are rarely used. The applicability of ES and BA in calibration is studied in terms of efficiency and accuracy.

The remainder of this paper is organized as follows. Section 2 demonstrates the components of the microscopic traffic simulator parameter calibration method, including the identification of parameters, brief introduction of BNN, and application of HA. Section 3 presents the application and effectiveness of the calibration method combining BNN and HA through a real case study in the TransModeler simulator. Section 4 discusses the results of the case study. Section 5 is the conclusion.

2. Calibration Method of Microscopic Traffic Simulator Parameters

The proposed parameter calibration method consists of three parts: identifying needed parameters to be calibrated, training the BNN model, and applying HA to find the optimal calibrated parameter values. The specific process of the method is shown in Figure 1.

2.1. Identifying Parameters for Calibration. The first step of calibrating parameter values in a traffic simulator is identifying needed parameters before demonstrating the calibration methodology, so a two-step procedure to identify parameters is proposed. Sensitivity analysis is used to verify whether the selected parameters significantly affect the simulation results, and a range test is conducted to know whether the field measurements are in the numerical range of simulator outputs.

2.1.1. Sensitivity Analysis. Generally speaking, based on experience, the initial selection of the parameters to be calibrated can be completed. Then, it is necessary to verify whether the selected parameters significantly affect the simulation results. This process is called sensitivity analysis. The maximum and minimum values are taken for a selected parameter, and the other parameters are kept at their default settings (including the random seed) to obtain the simulation results. Next, the default values of all parameters are kept, and the random seed is randomly used to obtain its effect.

Another role of sensitivity analysis is to explore whether the effect caused by the changes of parameters to be calibrated is significant compared with the effect caused by the random seed in the simulation outputs. If the effect caused by the former is significant, it is essential to do calibration to find the optimum parameter values.

When it comes to specific processes, an example road network is built in the microscopic traffic simulator. First, for a parameter, or called the ith parameter, two simulations are conducted, where the ith parameter uses the maximum and minimum values, respectively, according to the recommendation document, and other parameters take the simulator's

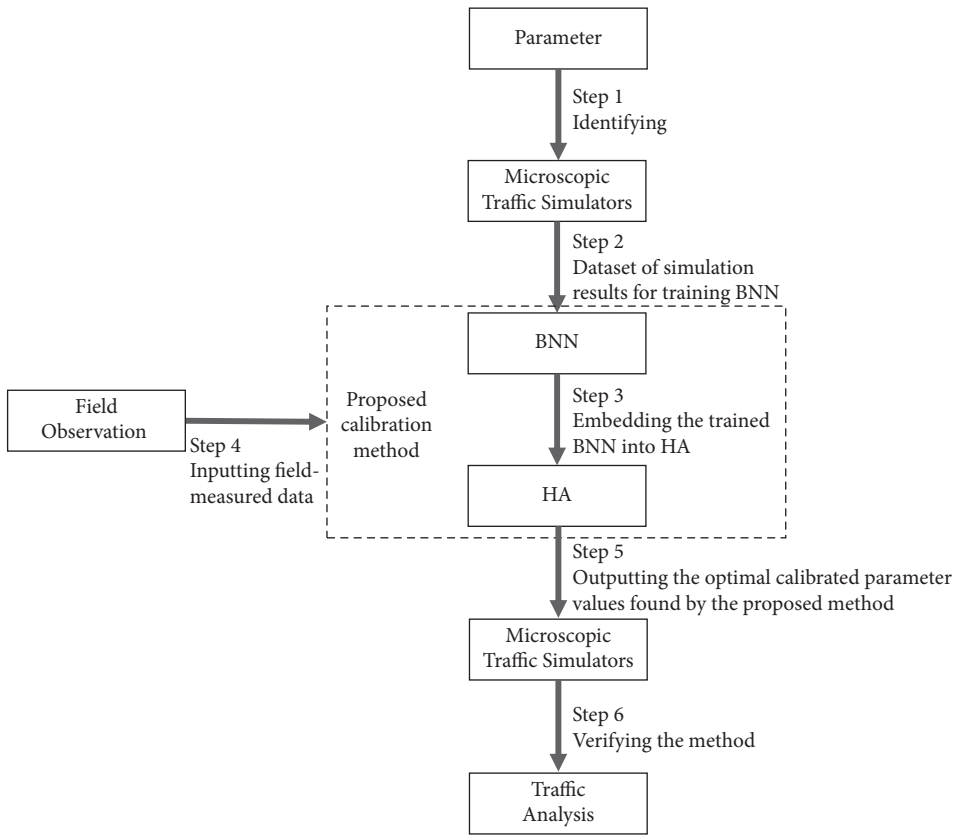


Figure 1: Process of the parameter calibration method.

default settings. The difference between the results of the two simulations is denoted as ΔR_i. Afterward, all parameters are set as default. Multiple simulations with randomly generated seeds are performed to obtain the effect caused by the random seed. And ΔR denotes the difference between the minimum and the maximum simulation results. If $\Delta R_i > \Delta R$, it can be concluded that the influence of the ith parameter on results is greater than the influence of the random seeds, so the ith parameter needs calibration.

2.1.2. Range Test. The purpose of the range test is to ensure that the range of the simulation results covers the field-measured data. According to the recommendation document, multiple sets of parameter values within the recommended range are randomly generated. For a set of parameter values, they are inputted into the traffic simulator and the simulation is run several times to calculate the average of the simulation results to reduce randomness. And the same process is repeated for all sets to acquire the range of the simulation results. Eventually, we examine whether the field-measured data are within the range of the simulation results. If not, extra revision to the parameters' range would be required [42].

2.2. Bayesian Neural Network. BNN is a kind of neural network. Different from other neural networks, the weight in BNN replaces a certain value with a probability distribution, as shown in Figure 2. Uncertainty in weights caused by probability distributions can bring benefits of regularization and richer prediction results with uncertainty. There are two kinds of uncertainty in BNN: epistemic uncertainty and aleatoric uncertainty [43]. Epistemic uncertainty derived from uncertainty in weights can bring prediction results with uncertainty. It can be reduced by enough training data. And aleatoric uncertainty is derived from the inherent noise in the data. The following section only demonstrates the main structure of BNN. For more technical details about BNN, the readers can refer to Reference [44].

A neural network can be considered as a probabilistic model $P(y|x, w)$ (x is the input vector, y is the possible output vector, and w is denoted as the weight). For a training data set $D = (x, y)$, $P(w|D) \propto P(D|w)P(w)$ can be derived based on the Bayesian theory. In BNN, regularization and avoidance of overfitting are achieved by setting prior distributions on w. And maximizing $P(w|D)$ gives the maximum a posteriori (MAP) weights w^{MAP}:

$$\begin{aligned} \mathrm{w}^{MAP} &= \arg\max_{\mathrm{w}} \log P(w|D), \\ &= \arg\max_{\mathrm{w}} \log P(D|w) + \log P(w). \end{aligned} \tag{1}$$

Because Bayesian posterior weight distribution $P(w|D)$ is difficult to calculate in a neural network, using a variational distribution to approximate $P(w|D)$ has become the solution. The variational inference tries to minimize the Kullback–Leibler (KL) divergence between a variational distribution on the weights $q(w|\theta)$ and the true $P(w|D)$ by finding the best parameter θ':

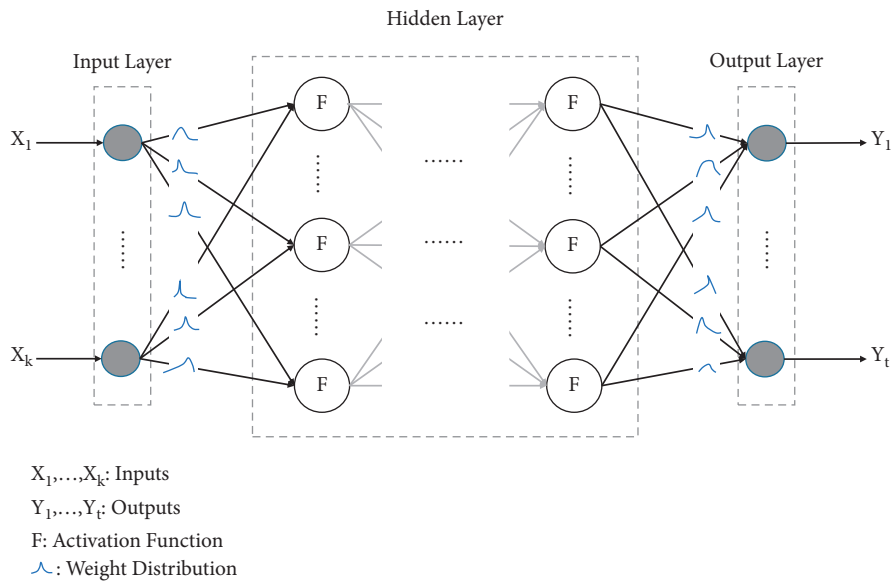


Figure 2: Multilayer Bayesian neural network structure.

$$\begin{aligned}\theta' &= \arg\min_{\theta} \mathrm{KL}[q(w|\theta)\|P(w|D)],\\ &= \arg\min_{\theta} \int q(w|\theta)\log\frac{q(w|\theta)}{P(w|D)}dw,\\ &= \arg\min_{\theta} \int q(w|\theta)\log\frac{q(w|\theta)}{P(D|w)P(w)}dw,\\ &= \arg\min_{\theta} \mathrm{KL}[q(w|\theta)\|P(w)] - E_{q(w|\theta)}[\log P(D|w)].\end{aligned} \tag{2}$$

The optimization objective function, which is also called the cost function and the variational free energy, can be defined as follows:

$$\mathcal{F}(D,\theta) = \mathrm{KL}[q(w|\theta)\|P(w)] - E_{q(w|\theta)}[\log P(D|w)]. \tag{3}$$

The first item in equation (3) is the complexity cost, which measures the KL divergence between $q(w|\theta)$ and prior probability distribution $P(w)$. Another item in equation (3) is the likelihood cost. Equation (3) can be rewritten as follows:

$$\begin{aligned}\mathcal{F}(D,\theta) = {} & E_{q(w|\theta)}[\log q(w|\theta)] - E_{q(w|\theta)}[\log P(w)]\\ & - E_{q(w|\theta)}[\log P(D|w)].\end{aligned} \tag{4}$$

Nevertheless, it is still hard to calculate equation (4). Using the Monte Carlo method, equation (4) can be approximated as follows:

$$\mathcal{F}(D,\theta) \approx \sum_{i=1}^{n} \log q\left(w^{(i)}|\theta\right) - \log P\left(w^{(i)}\right) - \log P\left(D|w^{(i)}\right), \tag{5}$$

where $w^{(i)}$ is the ith Monte Carlo sample taken from $q(w^{(i)}|\theta)$.

We assume that the variational distribution is Gaussian, controlled by the distribution mean μ and the distribution standard deviation σ. For numeric stability, a new parameter ρ is introduced so that $\sigma = \log(1+e^{\rho})$ can guarantee the nonnegativity of σ. Thus, (μ,ρ) constitutes the parameter θ of the variational distribution. For the weights w in the neural network, we use μ and ρ to control the distributions of weights, which means that the number of parameters to be learned is doubled compared with other neural networks.

In addition, a reparameterization trick is necessary. The transformed form of weights is $w = \mu + \sigma \circ \varepsilon = \mu + \log(1+e^{\rho}) \circ \varepsilon$, where $\circ$ is point-wise multiplication and $\varepsilon \sim N(0,I)$ is a sampled parameter-free noise. For the prior probability, a scale combination of two Gaussians is considered:

$$P(w) = \prod_{j} \pi N\left(w_j|0,\sigma_1^2\right) + (1-\pi)N\left(w_j|0,\sigma_2^2\right), \tag{6}$$

where w_j denotes the jth weight in the BNN, π and σ_k are hyperparameters, and $N(w_j|0,\sigma_k^2)$ denotes the Gaussian density with variance σ_k at w_j $(k=1,2)$.

Training BNN consists of two parts: forward iteration and backward iteration. In the forward iteration, a sample drawn from the variational distribution is used to calculate the approximate optimization objective function according to equation (5). And in the backward iteration, gradients of μ and σ are obtained to update their values by an optimizer.

In the detailed process of the BNN model training, parameter values to be calibrated are set as the inputs, and traffic simulation results from the simulator are the outputs. Therefore, the objective of training the BNN is to form the ability to predict simulation results based on inputs from traffic simulator parameter values without real simulation.

The data set for training the BNN is derived from multiple runs of the simulation. First, a large number of sets of parameter values are randomly generated. For each set,

the parameter values to be calibrated are inputted into the traffic simulator. And we simulate several times to obtain an averaged output. A set of parameter values and the corresponding average output forms a piece of training data. For different parameter value sets, the same process is repeated. Finally, numerous training data are acquired for training the BNN, validating the BNN, and testing the BNN.

2.3. Heuristic Algorithms for Calibration. The calibration problem inherently seeks and optimizes specific parameter values with the minimum difference between field-measured values and simulation output values. By using HA, we search for the values of the parameters in the microscopic traffic simulator to minimize the difference between field-measured values and the predicted simulation results from BNN. In the process of the HA, each individual in HA represents a set of parameter values to be calibrated. In multiple iterations, each individual constantly changes according to the rules of the algorithm. BNN is integrated into HA to output the simulation results of each individual. The mean squared error (MSE), also called the fitness value in the HA, is selected as an indicator to measure the difference between field-measured values and the BNN outputs. By minimizing the MSE, HA can find the set of parameter values (the optimal individual) to be calibrated, which is closest to the field situation.

After performing three HAs (GA, ES, and BA), the calibrated parameter values can be further used. Based on the calibrated parameters of microscopic traffic simulation simulators, the consistency between simulation and reality can be verified, and subsequent traffic analysis can be performed.

Before introducing the HA, the common notations are shown in Table 1.

2.3.1. Genetic Algorithm. The main point of GA is that each set of random parameter value combinations (initial population) has its own unique chromosome coding method. Before each iteration, some individuals in the population are randomly eliminated according to certain rules, and the performance of most of the eliminated individuals is poor (i.e., the fitness value is high in the case), and then the remaining individuals continue to reproduce. Under the joint action of elimination, reproduction, and mutation, after many iterations, the whole population will eventually gather at the optimal solution, which means they all have similar chromosome coding methods. The inspiration of GA comes from the law of survival of the fittest in the animal population. The closer the gene is to the optimal solution, the more likely the individual will survive. After several generations, the whole population will continue to adapt to the environment. Therefore, the algorithm has developed certain evolutionary advantages.

In GA, the input parameter set corresponding to each individual in the population corresponds to a chromosome-like code through some coding method with a fitness value. At the start of GA, each individual in the population will get a randomly generated chromosome code. In each iteration, some chromosomes should be selected randomly according to some rules, and others should be eliminated. The smaller the fitness value of each individual (this study aims to find the minimum fitness value), the greater the probability of being selected. The new individuals in the population are supplemented by the following ways: (1) pairing the selected chromosomes to generate new individuals between the two, and (2) random variation of some existing individuals, which changes a random part of the chromosome code into a new individual. The process repeats until the maximum iterations. Finally, the optimal chromosome code searched by the population can be restored according to the coding method to get the value of each parameter to be calibrated.

The following describes the detailed steps of GA:

Step 0. Initialize: Let $k = 1$. Assign random values to P'_{1i} within the value range ($i = 1, \ldots, \beta'_1$). Set $F' = \infty$ and $G' = 0$.

Step 1. Introduce the trained BNN model: For $i = 1, \ldots, \beta'_k$, input P'_{ki} into the BNN model to obtain predicted outputs X_{ki}, respectively.

Step 2. Update the best fitness value and the optimal parameter values: For each individual i, calculate $F'_{ki} = \sum_{j=1}^{\gamma'} (X'_{kij} - X'_{0j})^2/\gamma'$. If $\min\{F'_{ki}\} < F'$, then let $F' = \min\{F'_{ki}\}$ and $G' = \arg\min_{P'_{ki}}\{F'_{ki}\}$.

Step 3. Determine the probability of selecting the particle to enter the next iteration according to the fitness value of each individual, and generate the selected population.

Step 4. From the selected population, the individuals with the cross-ratio q are randomly selected, and their position coordinates are coded with the appropriate coding method, and the new offspring individuals are generated by pairing.

Step 5. Among the new individuals, the individuals with mutation rate r are selected randomly, and the chromosome code is changed. The selected individuals, the newly generated offspring, and the mutated offspring form a new species group.

Step 6. Examine stopping criteria: If $k < N'$, let $k = k + 1$ and continue from Step 1. Otherwise, stop after running Step 2.

2.3.2. Evolutionary Strategy. The idea of ES is similar to that of GA, both of which are derived from biological evolution. In ES, each individual is also coded with a different chromosome. However, unlike GA which uses binary to code chromosomes, ES directly uses real values to code chromosomes. Through multiple iterations of reproduction, mutation, and selection, the entire population will converge on the optimal solution.

In ES, the input parameter set corresponding to each individual in the population is set to the chromosome encoding value. Each individual is given a randomly generated set of parameter values in population initialization. In each iteration, the parameter value of each individual will produce small, random, and unbiased mutations based on

Table 1: Common notations.

Notation	Meaning
α'	The number of parameters for calibration
γ'	The number of observed links
β'_k	The population size of the whole population in the kth iteration
F'_{ki}	The current fitness of the ith individual in the kth iteration
F'	The best fitness value (MSE) for all individuals in the population
N'	The maximum number of iterations
$P'_{ki} = (P'_{ki1}, P'_{ki2}, \ldots, P'_{ki\alpha'})$	The parameter group of the ith individual in the kth iteration including α' parameters
$X'_0 = (X'_{01}, X'_{02}, \ldots, X'_{0\gamma'})$	The set of field-measured data
$G' = (G'_1, G'_2, \ldots, G'_{\alpha'})$	The parameter group associated with F'
$X_{ki} = (X_{ki1}, X_{ki2}, \ldots, X_{ki\gamma'})$	The BNN outputs of the ith individual in the kth iteration

an adaptive Gaussian distribution to get offspring [45]. For each individual, the fitness value is calculated separately for the parent and offspring, and then the individual with the smaller fitness value is inherited. The process repeats until the maximum iterations. Eventually, the parameter values of the individual with the smallest fitness are the optimal parameter values to be calibrated.

The following describes the detailed steps of ES:

Step 0. Initialize: Let $k = 1$. Assign random values to P'_{1i} within the value range $(i = 1, \ldots, \beta'_1)$. Set $F' = \infty$ and $G' = 0$.

Step 1. Introduce the trained BNN model: For $i = 1, \ldots, \beta'_k$, input P'_{ki} into the BNN model to obtain predicted outputs X_{ki}, respectively.

Step 2. Update the best fitness value and the optimal parameter values: For each individual i, calculate $F'_{ki} = \sum_{j=1}^{\gamma'} (X'_{kij} - X'_{0j})^2/\gamma'$. If $\min\{F'_{ki}\} < F'$, then let $F' = \min\{F'_{ki}\}$ and $G' = \arg\min_{P'_{ki}}\{F'_{ki}\}$.

Step 3. Mutation: Add a random number that follows an adaptive Gaussian distribution with zero mean affected by parameter variance δ_k to the parameter value of each parent individual in the population to obtain a new offspring.

Step 4. Selection: For $i = 1, \ldots, \beta'_k$, calculate and compare the fitness values of the ith individual's parent and offspring. Individuals with smaller fitness values are chosen to keep.

Step 5. Examine stopping criteria: If $k < N'$, let $k = k + 1$ and continue from Step 1. Otherwise, stop after running Step 2.

2.3.3. Bat Algorithm. BA simulates the natural process of bat predation using ultrasound. BA has the characteristics of rapid convergence in the early search stage, which reflects the efficiency of the algorithm. The principle of BA is to map each individual in a group of bats to a randomly generated feasible solution and use the process of searching for prey and movement of bats to conduct simulated search and optimization, respectively. The evaluation of the position of the bat is based on the fitness value.

Bats find food by continuously adapting their sound waves during predation, and BA is no exception. First, the bat performs a global search with louder sound waves and lower pulse frequencies. As the distance to the prey gets closer, the loudness of the sound waves decreases and the pulse frequency increases. During predation, the frequency of bats' emitted pulses and the pulse emission rate are automatically adjusted according to the distance to the prey [46, 47]. The process repeats until the maximum iterations. Finally, the feasible solutions corresponding to the bat individual with the smallest fitness value are the optimal values of parameters to be calibrated.

The unique notations of BA are shown in Table 2.

The following describes the detailed steps of BA:

Step 0. Initialize: Let $k = 1$. Assign random values to P'_{1i} within the value range $(i = 1, \ldots, \beta'_1)$. Set $F' = \infty$ and $G' = 0$. Initialize other parameters in BA.

Step 1. Introduce the trained BNN model: For $i = 1, \ldots, \beta'_k$, input P'_{ki} into the BNN model to obtain predicted outputs X_{ki}, respectively.

Step 2. Update the best fitness value and the optimal parameter values: For each individual i, calculate $F'_{ki} = \sum_{j=1}^{\gamma'} (X'_{kij} - X'_{0j})^2/\gamma'$. If $\min\{F'_{ki}\} < F'$, then let $F' = \min\{F'_{ki}\}$ and $G' = \arg\min_{P'_{ki}}\{F'_{ki}\}$.

Step 3. Update according to the following equation: $f_i = \min\{f\} + (\max\{f\} - \min\{f\}) \cdot \beta$, $V^{t+1}_{ki} = V^t_{ki} + (X^t_{ki} - G') \cdot f_i$, $X^{t+1}_{ki} = X^t_{ki} + V^{t+1}_{ki}$, where β is a randomly generated variable and subject to a uniform distribution from 0 to 1.

Step 4. Generate a random number a. If $a > r^k_i$, $G' = G' + \varepsilon L^k$, where ε is a randomly generated variable and subject to a uniform distribution from −1 to 1.

Step 5. Generate a random number b. If $b < L^k_i$ and $F'_{ki} < F'$, use $r^{k+1}_i = \max\{r\} \cdot (1 - e^{-mk})$ and $L^{k+1}_i = n \cdot L^k_i$ to generate a new individual, where m and n are constants.

Step 6. Examine stopping criteria: If $k < N'$, let $k = k + 1$ and continue from Step 1. Otherwise, stop after running Step 2.

3. Case Study

3.1. Background. The field-measured data used for the case study are collected from three highway links of the North-South Elevated Road that are easily congested in Shanghai,

Table 2: Unique notations of BA.

Notation	Meaning
f_i	The frequency of the ith bat
v_i^k	The velocity of the ith bat in the kth iteration
r_i^k	The pulse emission rate of the ith bat in the kth iteration
L_i^k	The loudness of the ith bat in the kth iteration
L^k	The average loudness of the whole bat population in the kth iteration
$V_{ki} = (V_{ki1}, V_{ki2}, \ldots, V_{ki\alpha'})$	The velocity set of the ith bat in the kth iteration including α' parameters

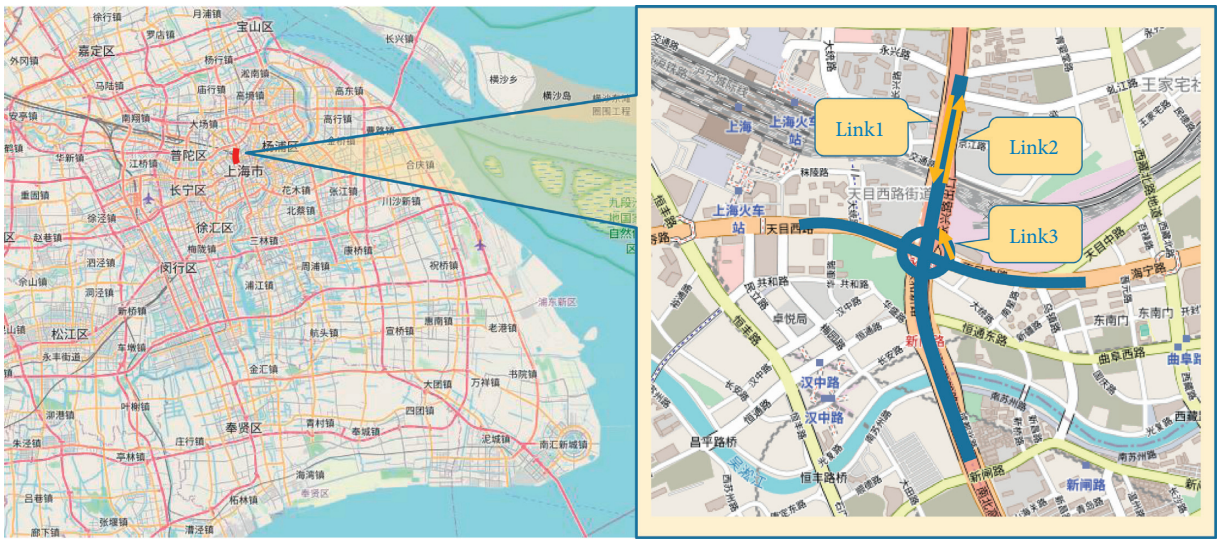


Figure 3: The geographical location of the three highway links in Shanghai, China.

China. The total length of the links is 2 km as shown in Figure 3. The data include the time and speed of the vehicle passing through the monitoring point of the link and the license plate number. The data collection time is from 9 am to 10 am on May 28, 2018, and the data source is Shanghai Municipal Traffic Management Bureau. Although the highway links do not have traffic lights, ground-level roads and nearby ramps have. Therefore, information about related traffic control is further gathered. Network geometric features, vehicular characteristics, and traffic volume are also collected. All information is inputted into the foundation of the traffic simulation model. In this case, we need to calibrate the average traffic speed on highway links.

Simulation is carried out in a universal microscopic traffic simulator TransModeler to verify the application of the proposed method in calibrating the parameters. TransModeler has various parameters that affect the simulation results (i.e., driver behaviors, car-following model parameters, and so on). Some parameters are the same in different regions, so they do not need calibrating. Nevertheless, parameters concerning driver behavior are different in different regions, which may have a huge impact on the simulation output and need calibration.

3.2. Parameter Identification. Some parameters that may need to be calibrated are initially selected in TransModeler. First, the car-following models in TransModeler are the basic logic for realizing simulation, so it is necessary to select the parameters in the models for calibration. The car-following models that are universal for different vehicle types are shown in the following equations:

$$A_i'[t+\Delta t] = a' \frac{(V_i[t])^{b'}}{(D_{i,i-1}[t])^{c'}} (V_{i-1}[t] - V_i[t]) + \varepsilon_i, \quad (7)$$

$$A_i''[t+\Delta t] = a'' \frac{(V_i[t])^{b''}}{(D_{i,i-1}[t])^{c''}} (V_{i-1}[t] - V_i[t]) + \vartheta_i. \quad (8)$$

Among them, equation (7) is about acceleration and equation (8) is about deceleration. $A_i'[t+\Delta t]$ denotes acceleration of the ith vehicle at time $t+\Delta t$. $A_i''[t+\Delta t]$ denotes deceleration of the ith vehicle at time $t+\Delta t$. $V_i[t]$ denotes the speed of the ith vehicle at time t. $V_{i-1}[t]$ denotes the speed of the $i-1$th vehicle (the front vehicle of the ith vehicle) at time t. $D_{i,i-1}[t]$ denotes the distance between the ith vehicle and the $i-1$th vehicle at time t. ε_i and ϑ_i denote errors. Readers can refer to Reference [48–50] to know more about car-following models. Based on the above models, six parameters (a', b', c', a'', b'', and c'') to be calibrated are selected. Subsequently, some other parameters are also selected, including the critical value of car-following speed, car-following time headway, and transverse speed when a vehicle changes

lane, transverse speed when a vehicle is in a transition section, and the maximum look-ahead distance of the driver's visibility. They are denoted as P_1, P_2, P_3, P_4, and P_5, respectively.

In the sensitivity analysis, average traffic speed (u) and traffic volume (q) on each highway link are regarded as simulation results. Thus, ΔR in Section 2.1 consists of Δq and Δu in this case. The number of random seeds in TransModeler is 10. Figures 4 and 5 illustrate the results of sensitivity analysis concerning q and u, respectively. Among them, the error bars represent the effect caused by the random seed from 10 simulations. The results of the sensitivity analysis are obtained by selecting the maximum and minimum values of the corresponding parameters and using the default values for other parameters to run the simulation. Comprehensively, Δq and Δu of a', b', c', a'', b'', and c'' are much larger than those of other parameters. Thus, a', b', c', a'', b'', and c'' are selected for calibration.

In the range test, 460 sets of values of 6 selected parameters are generated at random within the corresponding recommended ranges from the instructions of TransModeler. For each set, we run 10 simulations and take the average value of the simulation speed as the results. The results of the range test verify that the field-measured speeds are within the range of the distribution of simulation speeds.

3.3. Building BNN. The same data set as the range test is used for building the BNN model. The data set is randomly divided into a training set, a validation set, and a test set according to the ratio of 8 : 1 : 1. Moreover, the basic framework of the BNN model is implemented based on the logic introduced in the previous subsection, and the BNN model is built based on Python and its neural network API, Keras. The trained BNN model is evaluated by the indicators mean absolute error (MAE), MSE, root mean squared error (RMSE), and mean absolute percentage error (MAPE). After the training and the evaluation of multiple BNN models, we comprehensively selected the model with the lowest error on the validation set to find the relationship between the simulation results and the parameters to be calibrated as close as possible. The selected BNN model has two hidden layers, each with 23 neurons. Rectified linear unit (ReLU) is selected as the neuron activation function. The values of hyperparameters π, σ_1, and σ_2 are 0.5, 1.5, and 0.1, respectively.

The prediction ability of the BNN model is evaluated by the consistency of the simulation speeds in TransModeler and the predicted speeds from the BNN model based on the test set in the scatter plots with uncertainty, as shown in Figures 6–8. The horizontal axis of the scatter plot is the simulation speed, and the vertical axis is the predicted speed. The error bars of the data points in the vertical axis indicate the prediction range of the BNN model. If the predicted speeds are closer to the simulation speeds, in other words, the closer the data points are to the diagonal in the scatter plot, the better the model's predictive ability. And based on the test set, the indicators MAE, MSE, RMSE, and MAPE also demonstrate the predictive accuracy of the BNN model in Table 3. Although the BNN prediction error on Link 3 is slightly larger, especially on MAPE, which is due to the small denominator, the overall prediction accuracy is high.

3.4. Using HA for Seeking the Optimal Parameter Values. Based on the previously trained BNN model, three HAs are performed to seek the optimal parameter value set. HAs are implemented in Python. To reduce randomness, we run 10 times for each HA.

In this case, the parameters of GA are set as follows: $\beta_1' = 100$, $N' = 200$, $q = 0.8$, and $r = 0.55$ [33, 38]. The setting of these parameters significantly affects the performance of GA. The population size β_k' and the maximum iterations N' mainly affect the convergence of the final results. If the population size is too large or the maximum iterations are too small, the algorithm is difficult to converge. Additionally, the cross-ratio q and the mutation rate r control the renewal of the population. If the cross-ratio is too high, it is easy to destroy the existing favorable mode. If the crossover ratio is too low, the renewal of the population is inefficient. A low mutation rate will rapidly reduce the diversity of the population, which can lead to an irreparable loss of effective genes. However, if the mutation rate is too high, the probability of the high-order mode being destroyed is also very high. Therefore, in order to ensure the efficiency and accuracy of GA, it is necessary to determine the reasonable population size, generation size, cross-ratio, and mutation rate.

Based on the actual situation of this case study, the parameters of ES take the same values as GA: $\beta_1' = 100$, $N' = 200$, $\delta_{\max} = 12$, and $\delta_{\min} = 1$. The effects of the population size β_k' and the maximum number of iterations N' in ES are similar to those in GA. Moreover, δ_k is calculated as in the following equation in each iteration:

$$\delta_k = \delta_{\min} + (\delta_{\max} - \delta_{\min}) \frac{k}{N'}, \tag{9}$$

δ_k controls the variation of the population. When running ES at the beginning, δ_k is relatively large to facilitate global search. The global search aims to find the region where the optimal parameter values are located as much as possible. Otherwise, the results tend to converge to the local optimal solution, especially for large-scale problems. With the increase in iterations, δ_k gradually becomes smaller for local search to get the optimal parameter values.

According to Reference [29, 47] and actual case background, the parameters of BA in this case are set as follows: $\beta_1' = 100$, $N' = 50$, $m = n = 0.9$, $f_{\max} = 1$, and $f_{\min} = 0$. Small population sizes will bring about a poor search effect. Nevertheless, when the size is large, the calculation complexity will increase and the efficiency will be reduced. Because BA converges quickly, the maximum number of iterations can be reduced compared with GA and ES. The determination of the maximum number of iterations also requires a trade-off. If the number of iterations is too small, the randomness will increase when the algorithm stops searching, and if the number of iterations is too large, the efficiency of solving will also be

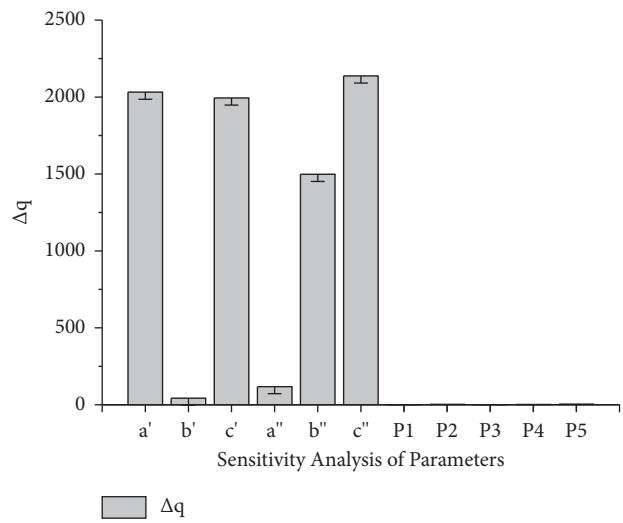


Figure 4: Sensitivity analysis results of traffic volume. The error bars represent the difference of traffic volume in 10 simulations with randomly generated seeds.

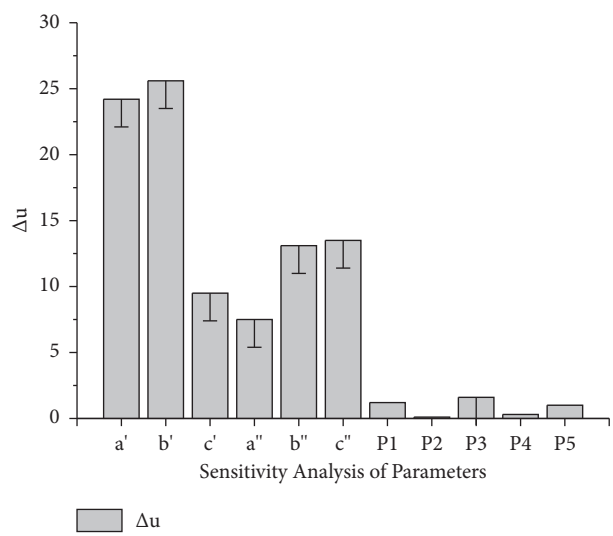


Figure 5: Sensitivity analysis results of average traffic speed. The error bars represent the difference of average traffic speed in 10 simulations with randomly generated seeds.

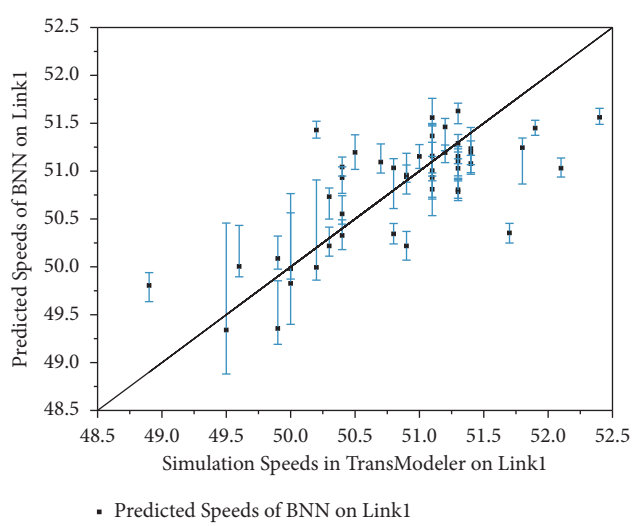


Figure 6: Scatter plot of simulation speeds and predicted speeds on Link 1. The error bars represent the 95% confidence interval of predicted speeds in 1000 predictions.

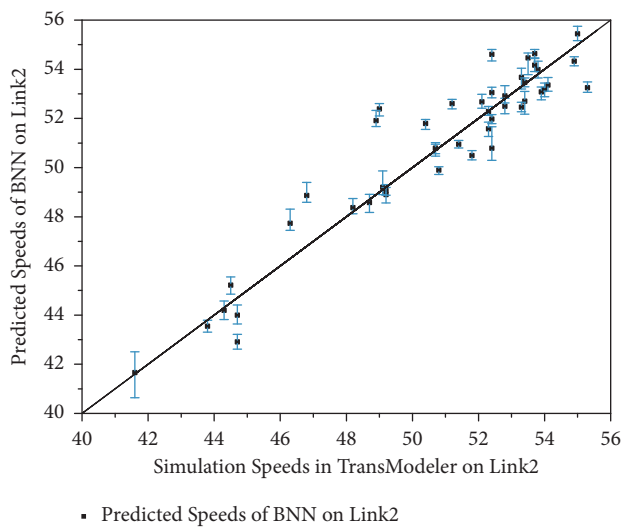


Figure 7: Scatter plot of simulation speeds and predicted speeds on Link 2. The error bars represent the 95% confidence interval of predicted speeds in 1000 predictions.

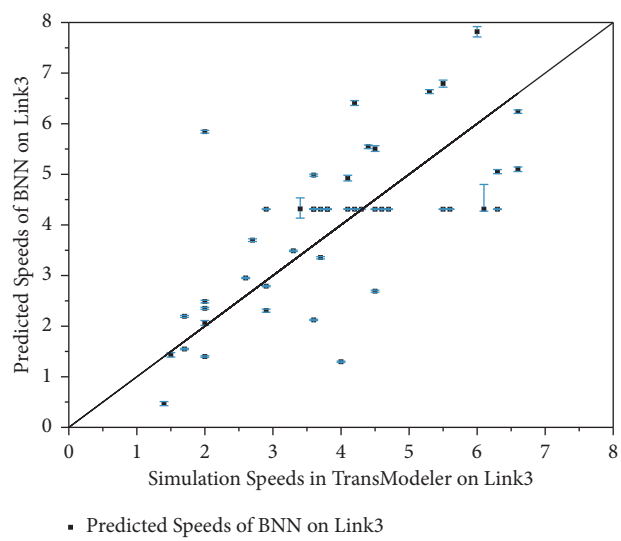


Figure 8: Scatter plot of simulation speeds and predicted speeds on Link 3. The error bars represent the 95% confidence interval of predicted speeds in 1000 predictions.

reduced. The values of the constants m and n and frequency also affect the efficiency and accuracy of the search. Therefore, it is necessary to determine the parameter values of BA.

4. Discussion

The previously proposed parameter calibration method combining ANN and GA (ANN-GA) is also introduced into the case study as a comparison [33, 39]. The ANN used has four hidden layers, each with 25 neurons. The dropout index is 0.2. The parameters of GA are the same as those in Section 3.4.

The change of the best fitness value in the iteration is shown in Figure 9. Table 4 shows the values of the calibrated parameter a', b', c', a'', b'', and c'', the MSE between the predicted result of neural network using the optimal parameter values found by HA and the field-measured data (the minimum optimal fitness value in 10 runs), and the average calculation time of HA in 10 runs. No matter from the perspective of MSE or the calculation time, BNN-BA has the highest accuracy and the highest calculation efficiency, followed by BNN-ES, and BNN-GA is the lowest. ANN-GA not only requires the longest average computation time but also has the largest MSE. This suggests that BA is an excellent heuristic optimization algorithm in this case.

The average calculation time of each HA is less than 31 seconds, which is almost negligible. It is worth noting that the average calculation time in table is just the time to run HA for calibration without the simulation. In this case, it takes 416 seconds to run a simulation of TransModeler. If the traditional calibration method is used, the simulation needs

Table 3: MAE, MSE, RMSE, and MAPE of the trained BNN model.

No.	MAE	MSE	RMSE	MAPE (%)
Link 1	0.369	0.234	0.484	0.7
Link 2	0.807	1.246	1.116	1.6
Link 3	0.924	1.451	1.205	25.9

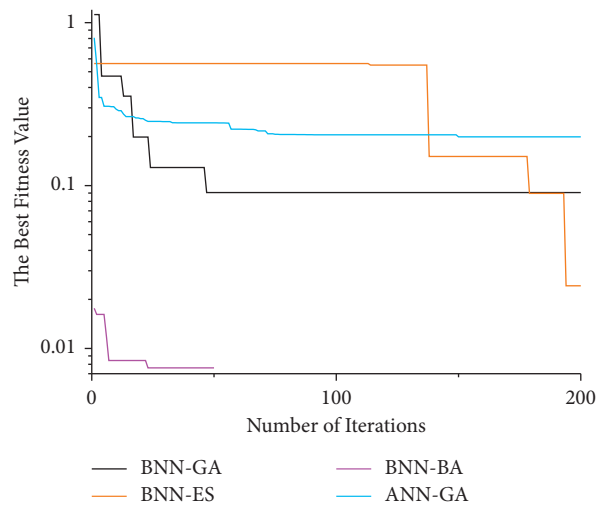


Figure 9: The best fitness value of the four methods.

Table 4: Results of the calibration methods.

Method	a'	b'	c'	a''	b''	c''	MSE (optimal fitness value)	Average calculation time (seconds)
BNN-GA	3.75	−1.35	−2.70	0.38	2.32	−0.03	0.091	18.898
BNN-ES	5.00	−2.14	−2.21	0.22	1.74	2.51	0.018	16.467
BNN-BA	4.73	−2.25	−1.62	−1.10	−0.62	−0.40	0.001	8.054
ANN-GA	4.99	−0.58	−2.99	−0.27	−2.99	−2.99	0.199	30.298

to be run many times, and the calibration time will be several times longer than 416 s. If the traditional GA-based calibration method is performed, 20000 simulations (the population size multiplied by the number of iterations) need to be run during the calibration based on the same parameters of GA. Even if the traditional calibration method using high-efficiency BA is performed, 5000 simulations still need to be run during the calibration based on the same parameters. Thus, the calibration time of the traditional method is much longer than that of the proposed method. Even if the proposed method considers the number of simulations for obtaining the training dataset, the total number of simulations in this method is still much lower than that of the traditional method. Furthermore, it is only necessary to simulate in order to obtain data for training the BNN model before the calibration process. Therefore, the time saved by not running the simulation during the calibration process is the biggest benefit of the proposed method.

Apart from this, although the number of highway links in the previous case study is only three, considering that the parameters to be calibrated and the road network scale are independent, the impact of the road network scale on the calculation time for running HA to calibrate is limited. If the influence exists, it would likely to be derived from performing the trained BNN model to predict the simulation results and updating the fitness value in HA. However, predicting and updating only need simple calculations, which take very little time. In a word, the shortened expected calculation time almost without the influence of network scale is also the benefit of the proposed method.

Figure 10 illustrates the average speeds of highway links based on BNN-GA, including the output results of TransModeler using the default parameter values (uncalibrated), the calibrated output results of TransModeler using the optimal parameter values found by BNN-GA (calibrated by BNN-GA), and the BNN predicted results using the optimal parameter values found by the combination of BNN-GA (predicted by BNN). The points in Figure 10 indicate the average speed measured in the field. Figures 11–13 show the calibration results of BNN-ES, BNN-BA, and ANN-GA, respectively, in the same way. It is obvious that the calibrated results and the BNN predicted results have close similarity, which verifies that the trained BNN model can replace TransModeler to get simulation outputs, not only in accuracy but also in

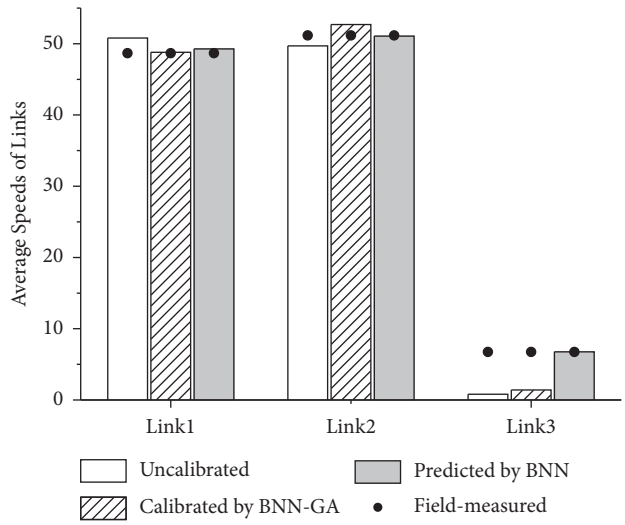


Figure 10: Calibration results using BNN-GA.

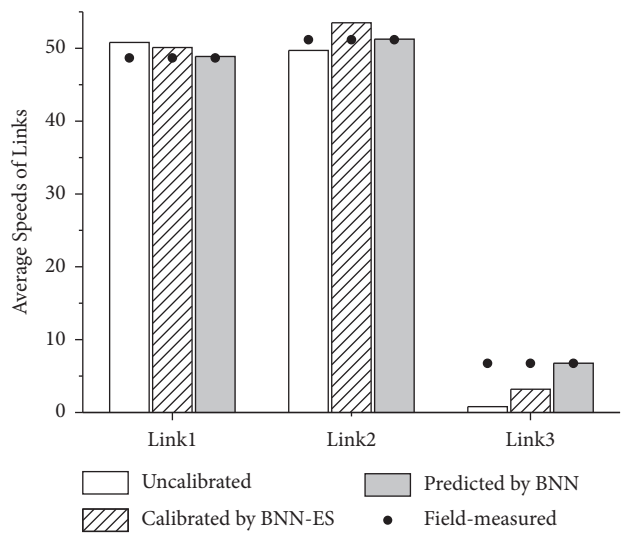


Figure 11: Calibration results using BNN-ES.

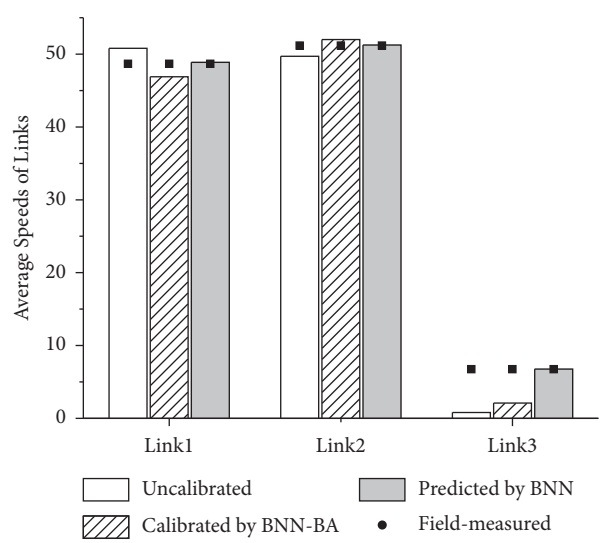


Figure 12: Calibration results using BNN-BA.

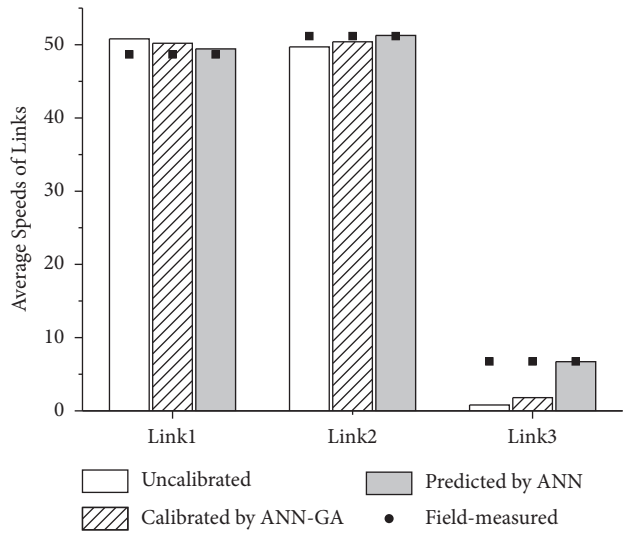


Figure 13: Calibration results using ANN-GA.

Table 5: Square error of three links.

No.	Uncalibrated	Calibrated by BNN-GA	Calibrated by BNN-ES	Calibrated by BNN-BA	Calibrated by ANN-GA
Link 1	4.558	0.018	2.059	3.115	2.356
Link 2	2.148	2.355	5.450	0.696	0.586
Link 3	35.286	28.518	12.533	21.531	24.406
Mean (MSE)	13.997	10.297	6.681	8.448	9.116

computationally efficiency. And the similarity between the BNN predicted results and the field-measured data suggests that inputting the parameter values found by the combination of BNN and HA into the BNN model can perfectly estimate actual road conditions.

Table 5 presents the square error of the default output results of TransModeler (uncalibrated) and the calibrated output results of TransModeler using the optimal parameter values found by the combination of neural network and HA, respectively.

Although there is some error in the simulation results predicted by BNN, the error is within acceptable limits compared with the improvement in the efficiency of the calibration calculation. No matter which HA is used for calibration, the MSE is lower than that using default and uncalibrated parameter values. This verifies the validity of the parameter calibration method proposed in this paper. Two calibration methods using GA (BNN-GA and ANN-GA) have poor performance in terms of MSE and the average calculation time for calibration. Considering the trade-off between MSE and calculation time, BNN-ES and BNN-BA perform well. BNN-ES has the lowest MSE (6.681) after calibration, while BNN-BA requires a shorter calculation time (8.054S) and has a relatively low MSE (8.448).

5. Conclusions

In this paper, a new parameter calibration method for microscopic traffic simulators combining BNN and HA is proposed. The objective of the BNN is to quickly predict simulation results with uncertainty based on inputs of traffic simulator parameter values without real simulation. Based on the trained BNN model, the purpose of running HA is to search for the optimal values of the parameters to be calibrated in the traffic simulator to minimize the difference between field-measured values and the predicted simulation results from BNN without simulation. The combination of BNN and HA avoids running the simulator repeatedly during the calibration, which can significantly decrease the computation time of calibration. The research innovation lies not only in the huge improvement of calibration efficiency but also in which the method is universal for microscopic traffic simulators.

Based on a real case, the calibration method combining BNN and HA is applied to calibrate parameter values in TransModeler, which also validates the proposed method in error and efficiency. The result shows that the BNN model captures the uncertainty of the simulation and has high predictive accuracy. And it takes almost no time to seek the parameter value set with a small MSE by using the combination of BNN and HA. When comparing the average calculation time, BNN-BA has the highest calculation efficiency, followed by BNN-ES. Moreover, in terms of the MSE between the calibrated output results and the field-measured data, BNN-ES performs best, BNN-BA performs moderately, and BNN-GA performs worst. Therefore, considering that the total calibration accuracy is high in the balance between calibration efficiency and calibration error, BNN-ES and BNN-BA have their own advantages to calibrate parameters in microscopic traffic simulators compared with BNN-GA and the existing ANN-GA.

The proposed method can quickly obtain the calibrated parameter values after training the BNN, which facilitates the simulation to be closer to the actual situation. The parameter value found is suitable for a certain area where the case is located. The BNN model in the proposed method can also be used by traffic authorities to obtain predictive real-time traffic conditions without running microscopic traffic simulators.

Future research can focus on how to optimize the hyperparameters in the BNN and establish a more reliable variational distribution to better approximate the Bayesian posterior weight distribution and predict the distribution of simulation outputs. Furthermore, based on the inherent characteristics of HA, the results inevitably end at a locally optimal solution. It may be meaningful to further improve HA for calibration to find the global optimal solution rather than the locally optimal solution.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Project of Shanghai Science and Technology Innovation Action Plan (grant no. 20511101800) and National Natural Science Foundation of China (grant no. 52002243).

References

[1] W. Wu, R. Sun, A. Ni, Z. Liang, and H. Jia, "Simulation and evaluation of speed and lane-changing advisory of CAVS at work zones in heterogeneous traffic flow," *International Journal of Modern Physics B*, vol. 34, no. 21, Article ID 2050201, 2020.

[2] M. Yanai, K. Abe, T. Yamada, H. Fujii, and S. Yoshimura, "Cluster analysis for a series of microscopic traffic simulation results," *Journal of Advanced Simulation in Science and Engineering*, vol. 4, no. 1, pp. 78–98, 2018.

[3] J. Barceló, "Fundamentals of traffic simulation," *Springer-Verlag GmbH*, vol. 145, no. 145, 2010.

[4] H. Yang, S. Han, and X. Chen, *Parameter Calibration and Application for the Vissim Simulation Model*, Urban Transport of China, Beijing, China, 2006.

[5] Y. Kang, M. Zhai, H. Zhang, and C. Zhang, "Parameter calibration of VISSIM simulation model based on the orthogonal experiment," in *Proceedings of the Tenth International Conference of Chinese Transportation Professionals*, Beijing, China, 2010.

[6] M. Farzaneh and H. Rakha, "Procedures for calibrating TRANSYT platoon dispersion model," *Journal of Transportation Engineering*, vol. 132, no. 7, pp. 548–554, 2006.

[7] A. Agarwal, D. Ziemke, and K. Nagel, "Calibration of choice model parameters in a transport scenario with heterogeneous traffic conditions and income dependency," *Transportation letters*, vol. 12, no. 7, pp. 441–450, 2020.

[8] C. Zhang, C. Osorio, and G. Flötteröd, "Efficient calibration techniques for large-scale traffic simulators," *Transportation Research Part B: Methodological*, vol. 97, pp. 214–239, 2017.

[9] A. U. Z. Patwary, W. Huang, and H. K. Lo, "Metamodel-based calibration of large-scale multimodal microscopic traffic simulation," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102859, 2021.

[10] R.-L. Cheu, X. Jin, K.-C. Ng, Y.-L. Ng, and D. Srinivasan, "Calibration of FRESIM for Singapore expressway using genetic algorithm," *Journal of Transportation Engineering*, vol. 124, no. 6, pp. 526–535, 1998.

[11] S. J. Kim, W. Kim, and L. R. Rilett, "Calibration of Microsimulation Models Using Nonparametric Statistical Techniques," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1935, no. 1, pp. 111–119, 2005.

[12] B. Park and H. Qi, "Development and evaluation of a procedure for the calibration of simulation models," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1934, no. 1, pp. 208–217, 2005.

[13] S. Menneni, C. Sun, and P. Vortisch, "Microsimulation calibration using speed-flow relationships," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2088, no. 1, pp. 1–9, 2008.

[14] O. Giuffrè, A. Granà, M. L. Tumminello, and A. Sferlazza, "Estimation of passenger car equivalents for single-lane roundabouts using a microsimulation-based procedure," *Expert Systems with Applications*, vol. 79, pp. 333–347, 2017.

[15] S. Chiappone, O. Giuffrè, A. Granà, R. Mauro, and A. Sferlazza, "Traffic simulation models calibration using speed-density relationship: an automated procedure based on genetic algorithm," *Expert Systems with Applications*, vol. 44, pp. 147–155, 2016.

[16] R. Omrani and L. Kattan, "Concurrent estimation of origin-destination flows and calibration of microscopic traffic simulation parameters in a high-performance computing cluster," *Journal of Transportation Engineering, Part A: Systems*, vol. 144, no. 1, Article ID 04017068, 2018.

[17] T. Ma and B. Abdulhai, "Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1800, pp. 6–15, 2002.

[18] G. Amirjamshidi and M. J. Roorda, "Multi-objective calibration of traffic microsimulation models," *Transportation letters*, vol. 11, no. 6, pp. 311–319, 2019.

[19] K. Bhattacharyya, B. Maitra, and M. Boltze, "Calibration of micro-simulation model parameters for heterogeneous traffic using mode-specific performance measure," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 1, pp. 135–147, 2020.

[20] C. Wang and C. Xu, "On the effects of various measures of performance selections on simulation model calibration performance," *Journal of Advanced Transportation*, vol. 2018, pp. 1–16, 2018.

[21] M. Dubska, A. Herout, R. Juranek, and J. Sochor, "Fully automatic roadside camera calibration for traffic surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1162–1171, 2015.

[22] B. Kostic, L. Meschini, and G. Gentile, "Calibration of the demand structure for dynamic traffic assignment using flow and speed data: exploiting the advantage of distributed

computing in derivative-free optimization algorithms," *Transportation Research Procedia*, vol. 27, pp. 993–1000, 2017.

[23] D. K. Hale, C. Antoniou, M. Brackstone, D. Michalaka, A. T. Moreno, and K. Parikh, "Optimization-based assisted calibration of traffic simulation models," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 100–115, 2015.

[24] S. Oh, R. Seshadri, C. L. Azevedo, and M. E. Ben-Akiva, "Demand calibration of multimodal microscopic traffic simulation using weighted discrete SPSA," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 5, pp. 503–514, 2019.

[25] Q. Liu, L. Wu, W. Xiao, F. Wang, and L. Zhang, "A novel hybrid bat algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 73, pp. 67–82, 2018.

[26] R. M. Rizk-Allah and A. E. Hassanien, "New binary bat algorithm for solving 0-1 knapsack problem," *Complex & Intelligent Systems*, vol. 4, no. 1, pp. 31–53, 2018.

[27] X. S. Yang and A. Hossein Gandomi, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, no. 5, pp. 464–483, 2012.

[28] S. Srivastava and S. K. Sahana, "Application of bat algorithm for transport network design problem," *Applied Computational Intelligence and Soft Computing*, vol. 2019, Article ID 9864090, 12 pages, 2019.

[29] X. S. Yang and X. He, "Bat algorithm: literature review and applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, p. 141, 2013.

[30] A. D. Lidbe, A. M. Hainen, and S. L. Jones, "Comparative study of simulated annealing, tabu search, and the genetic algorithm for calibration of the microsimulation model," *Simulation*, vol. 93, no. 1, pp. 21–33, 2017.

[31] M. Yu and W. Fan, "Calibration of microscopic traffic simulation models using metaheuristic algorithms," *International Journal of Transportation Science and Technology*, vol. 6, no. 1, pp. 63–77, 2017.

[32] X. S. Yang, *A New Metaheuristic Bat-Inspired Algorithm*, Springer, Berlin, Germany, 2010.

[33] H. Shahrokhi Shahraki, C. Alecsandru, R. Maghsoudi, and L. Amador, "An efficient soft computing-based calibration method for microscopic simulation models," *Journal of Transportation Safety & Security*, vol. 10, no. 4, pp. 367–386, 2018.

[34] N. Dadashzadeh, M. Ergun, A. S. Kesten, and M. Zura, "Improving the calibration time of traffic simulation models using parallel computing technique," in *Proceedings of the IEEE MT-ITS2019—6th International Conference on Models and Technologies for Intelligent Transportation Systems*, June 2019.

[35] G. Xiao, R. Wang, C. Zhang, and A. Ni, "Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 22907–22925, 2020.

[36] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[37] I. Ištoka Otković, T. Tollazzi, and M. Šraml, "Calibration of microsimulation traffic model using neural network approach," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5965–5974, 2013.

[38] N. T. Ratrout, S. M. Rahman, and I. Reza, "Calibration of PARAMICS model: application of artificial intelligence-based approach," *Arabian Journal for Science and Engineering*, vol. 40, no. 12, pp. 3459–3468, 2015.

[39] Y. Liu, B. Zou, A. Ni, L. Gao, and C. Zhang, "Calibrating microscopic traffic simulators using machine learning and particle swarm optimization," *Transportation letters*, vol. 13, no. 4, pp. 295–307, 2020.

[40] M. Rahman, M. Chowdhury, T. Khan, and P. Bhavsar, "Improving the efficacy of car-following models with a new stochastic parameter estimation and calibration method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2687–2699, 2015.

[41] F. Abodo, A. Berthaume, S. Zitzow-Childs, and L. Bobadilla, "Strengthening the case for a bayesian approach to car-following model calibration and validation using probabilistic programming," ", IEEE, in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference–ITSC*Auckland, New Zealand, , 2019.

[42] P. Byungkyu and Q. Hongtu, "Microscopic simulation model calibration and validation for freeway work zone network - a case study of VISSIM," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, September 2006.

[43] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Proceedings of the 31st Conference on Neural Information Processing System*, Long Beach, CA, USA, 2017.

[44] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," *Computer Science*, 2015.

[45] N. Hansen, D. V. Arnold, and A. Auger, "Evolution strategies," in *Springer Handbook of Computational Intelligence*, pp. 871–898, Springer, Berlin, Germany, 2015.

[46] X. S. Yang and S. Xin, "Bat algorithm for multi-objective optimisation," *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, p. 267, 2011.

[47] A. H. Gandomi and X.-S. Yang, "Chaotic bat algorithm," *Journal of Computational Science*, vol. 5, no. 2, pp. 224–232, 2014.

[48] H. Subramanian, *Estimation of Car-Following Models*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.

[49] K. I. Ahmed, *Modeling Drivers' Acceleration and Lane Changing Behavior*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

[50] H. Farah and H. N. Koutsopoulos, "Do cooperative systems make drivers' car-following behavior safer?" *Transportation Research Part C: Emerging Technologies*, vol. 41, pp. 61–72, 2014.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5808206, 18 pages
https://doi.org/10.1155/2021/5808206

Review Article

Survey on Deep Learning-Based Marine Object Detection

Ruolan Zhang, Shaoxi Li, Guanfeng Ji, Xiuping Zhao, Jing Li, and Mingyang Pan

Navigation College, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Shaoxi Li; lishaoxi@dlmu.edu.cn

Received 17 July 2021; Accepted 4 October 2021; Published 25 November 2021

Academic Editor: Chunjia Han

We present a survey on marine object detection based on deep neural network approaches, which are state-of-the-art approaches for the development of autonomous ship navigation, maritime surveillance, shipping management, and other intelligent transportation system applications in the future. The fundamental task of maritime transportation surveillance and autonomous ship navigation is to construct a reachable visual perception system that requires high efficiency and high accuracy of marine object detection. Therefore, high-performance deep learning-based algorithms and high-quality marine-related datasets need to be summarized. This survey focuses on summarizing the methods and application scenarios of maritime object detection, analyzes the characteristics of different marine-related datasets, highlights the marine detection application of the YOLO series model, and also discusses the current limitations of object detection based on deep learning and possible breakthrough directions. The large-scale, multiscenario industrialized neural network training is an indispensable link to solve the practical application of marine object detection. A widely accepted and standardized large-scale marine object verification dataset should be proposed.

1. Introduction

Information technology and intelligent development have changed the operation mode and direction of many industries. Traditional maritime shipping industry also has gradually been advanced from digitization and informatization to intelligence [1]. As a major advance in machine learning over the last decades, the deep learning approach is becoming the most powerful technique for intelligent transportation system [2]. The deep learning methodologies are applied in various fields in the maritime industry such as ship classification, object detection, collision avoidance, risk perception, and anomaly detection. The main application directions can be summarized as maritime surveillance and autonomous ship navigation.

Currently, most of the research focuses on some aspects of the deep learning technique that performance much higher than humans; however, that technique is unable to complete complex tasks. So far, although seafarers have some limitations and many failure examples in the process of completing shipping transportation, humans are still the most reliable executors. Therefore, it is necessary to survey the applications of deep learning-based technologies in the maritime field to explore how computer vision replaces or even surpasses humans in a real-world application, especially the object detection task, which has exploded in recent years, and most of the evaluation index has already made great progress.

Humans perceive the external objects' size, brightness, color, and movement state through their eyes; 80% of human perception information (situation awareness) is obtained through vision. With the limitation of the seafarer's lookout at the ship bridge, the visual perception of the horizon cannot be done excluding the solar direction [3] and bad weather often affects the seafarer's judgment. Most collisions and grounding are due to wrong interpretation or disregard of improper lookout (COLREGS-1972) [4].

Computer vision is an interdisciplinary scientific field that obtains and completes a series of image information processing from digital images or videos [5]. From the perspective of engineering, it seeks to perceive, understand, and automate tasks that the human visual system does.

Visual perception is an information-based approach to understanding biological and artificial vision [6]. It refers to the process of organizing, identifying, and interpreting visual information in environmental expression and

understanding. According to this definition, the goal of computer vision is to express and understand the environment. The core issue of visual perception is to study how to organize the input image information, identify objects and scenes, explain the content of the image.

A number of surveys of general object detection have been published in recent years. Zou et al. [7] reviewed more than 400 papers on the development of object detection technology from 1998 to 2018. This survey includes historical milestone detectors, detection datasets, measurement methods, and the latest detection methods. This article also reviews some important detection applications, such as pedestrian detection, face detection, and text detection, and conducts an in-depth analysis of the challenges and technological improvements in recent years. Jiao et al. [8] analyzed the existing typical object detection model and methods and discussed how to construct an effective and efficient system architecture based on the current detection model. Wu et al. [9] systematically analyzed the existing deep learning-based object detection frameworks and organized the survey into three major parts: (i) detection components, (ii) learning strategies, and (iii) applications and benchmarks. This survey covers a variety of elements affecting detection performance, such as detector architectures, feature learning, proposal generation, and sampling strategies. Chen et al. [10] analyzed the characteristics of the imbalance problem in different kinds of deep detectors and experimentally compared the performance of some state-of-the-art solutions on the COCO benchmark. Qiao et al. [11] combined the visual perception tasks required for maritime surveillance with those required for intelligent ship navigation to form a marine computer vision-based situational awareness complex and investigated the key technologies they have in common. This review focuses on the ship detection by the ship's own equipment and does not include the influence of other possible objects and backgrounds at sea, as well as the problems that may arise in industrial applications.

Computer vision-based marine object detection, as one of the most fundamental and challenging issues in maritime intelligent transportation, has received great attention over the last decades. As shown in Figure 1, it indicates the increasing number of publications in marine object detection from 2012 to 2021 July, the growing number of papers that their title is associated with "marine object detection" and "deep learning" over the past decades. The three advancements of digital data collection, computing power, and algorithm have promoted this deep learning research and application boom in maritime fields [12–15].

In recent years, deep learning-based visual perception has been widely applied to autonomous ship navigation and maritime transportation surveillance for intelligent transportation systems (ITS). The survey articles in the maritime field-related applications of computer vision are as follows: Qiao et al. [11] summarized the progress made in four aspects: full scene parsing of an image, ship reidentification, ship tracking, and multimodal data fusion with different visual sensors. Prasad et al. [16] provided a comprehensive overview of various approaches of video processing for object detection in the maritime environment. It consists of three modules: horizon detection, static background subtraction, and foreground segmentation. Moniruzzaman et al. [17] described the use of deep learning for underwater imagery analysis, and deep learning architectures have been highlighted. Hashmani et al. [18] presented a survey on edge detection-based and machine learning-based marine horizon line detection; each study is presented with a recommendation for their suitability for a specific application in the marine environment.

Also, projection-based, region-based, hybrid, and artificial neural network (ANN) based methods for sea horizon detection have been discussed [19]. The researches of ANN methods in maritime surveillance made the horizon line detection easy, accurate, and robust. For optical remote sensing images applied in maritime, Li et al. [20] summarized the detection and classification of ship optical remote sensing images. Both methods were analyzed for traditional feature-designed methods and the deep convolutional neural networks (CNN).

The main difference between this paper and the above surveys is summarized as follows: (i) this paper only focuses on the task of deep learning-based object detection in computer vision. (ii) It analyzes the state-of-the-art of marine object detection in maritime surveillance, autonomous ship navigation, and other related applications. (iii) It analyzes and discusses the factors that affect the state-of-the-art solutions, especially the mainstream datasets and the milestone detectors.

As far as we know, the aim of this paper is to provide a survey of the most important approaches in the field of deep learning-based object detection for the maritime transportation system. This survey focuses on describing and analyzing deep learning-based marine object detection tasks. We contribute to the following:

Literature evaluation: we summarize the existing application scenarios of visual object detection in the maritime field. (2) Comparison of existing datasets. In practical engineering problems, big data plays an important role in realizing industrial applications. (3) Special emphasis on the role and development direction of visual detection in the autonomous ship navigation scenario. (4) Discussing the current limitations of object detection based on deep learning and possible breakthrough directions.

The rest of this paper is organized as follows. Section 2 highlights the state-of-the-art methods for general object detection. Section 3 introduces the application of object detection based on deep learning in various subdivisions of maritime affairs. The state-of-the-art backbone-based models and important datasets are described in Section 4.

2. State-of-the-Art of Object Detection

The definition of object detection is the task of detecting instances of targets of a certain class within an image or video.

Generally speaking, the detection task consists of two subtasks. One is the category information and probability of the target, and it is a classification task. The second is the specific location information of the target, which is a positioning task.

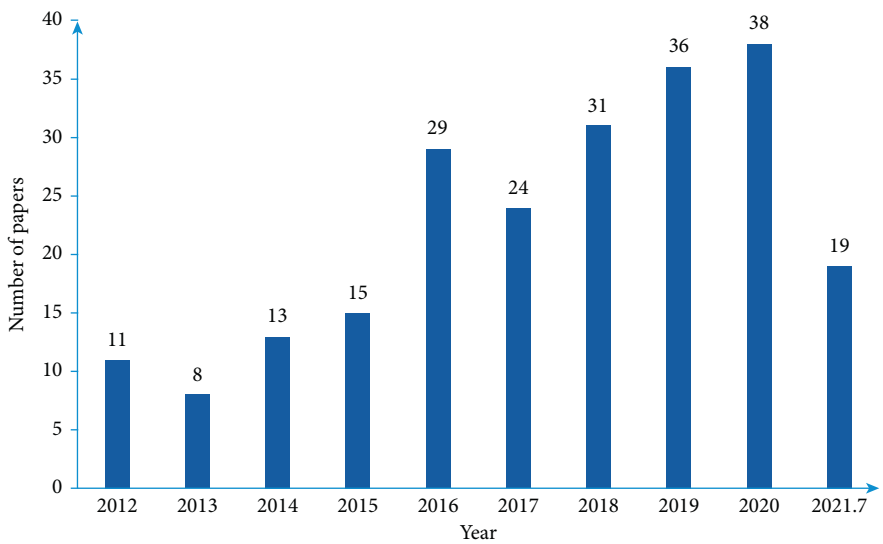


Figure 1: The increasing number of publications in marine object detection from 2012 to 2021 July (data from Google scholar advanced search).

As one of the most popular research fields of computer vision, object detection research prospered, which is the basic idea changed from traditional artificial feature design, shallow classifiers to deep neural network-based feature autonomous learning.

In the nondeep learning era, many tasks are not solved at once but require multiple steps, such as [21]. In the deep learning era, many tasks use the end-to-end framework, that is, input a picture and output the final result. The algorithm details and learning process are all completed through neural networks. It is particularly obvious in the field of object detection.

Under the deep learning architecture, whether it is a clear step-by-step process or the end-to-end method, the object detection algorithm must have three modules. The first is the selection of the detection window, the second is the extraction of image features, and the third is the design of the classifier.

As shown in Figure 2, the milestone of the neural network backbone and SOTA methods of object detection is listed in the timeline. 2012 was a critical point. Although CNN was proposed many years before, it was still hidden by other machine learning algorithms. After 2012, various neural networks and modules were combined, and deep learning-based methods suddenly left other methods behind. Deep learning-based application research in various fields can be widely carried out.

2.1. Review of Traditional Object Detection Methods. In 2001, [22, 23] proposed the Viola-Jones object detection framework. Based on the AdaBoost algorithm [24], the Viola-Jones framework uses Haar-like wavelet features and integral graph technology to perform face detection. This is the first detection method based on Haar + AdaBoost. It is also the first real-time framework for detection. Before the advent of deep learning technology, the Viola-Jones detector has always been the mainstream framework for face detection algorithms [25, 26].

Histogram of oriented gradient (HOG) [27] calculates the histogram not based on the color value but based on the gradient. It constructs the feature by calculating the gradient direction histogram of the local area of the image. HOG features combined with SVM classifiers have been widely used in image recognition, especially in pedestrian detection [28, 29]. Many further related researches have been presented, such as invariant histograms of oriented gradients (Ri-HOG) [30], which adopt annular spatial bins type cells and apply radial gradient transform (RGT) to attain gradient binning invariance for feature descriptors.

The DPM [31] algorithm adopts the detection ideas of improved HOG, SVM classifier, and sliding window. For the multiview problem of the target, it adopts the strategy of multicomponent. For the deformation problem of the target itself, it adopts the component model strategy of pictorial structure. DPM is a component-based detection method, which has strong robustness to the deformation of the target. At present, DPM has become the core of many classification, segmentation, pose estimation, and other deep learning-based algorithms [32–35].

In some specific application scenarios, object detection algorithms based on machine learning can still maintain good advantages. In [36], the image data were divided into smaller blocks and represented with a vector. These feature vectors are created by adding the subfeatures extracted from the color and texture properties of the images one after another. 99.62% classification success was achieved by using the Random Forest method. An average of 3.4 times acceleration was achieved by running each method on 1 master +4 workers clustering architecture on Apache Spark.

2.2. Deep Learning-Based Object Detection. CNN is one of the representative algorithms of deep learning [37]. It is the cornerstone of the current great success of deep learning, and it is a type of Feed forward Neural Networks (FNN) that includes convolution calculations and has a deep structure.

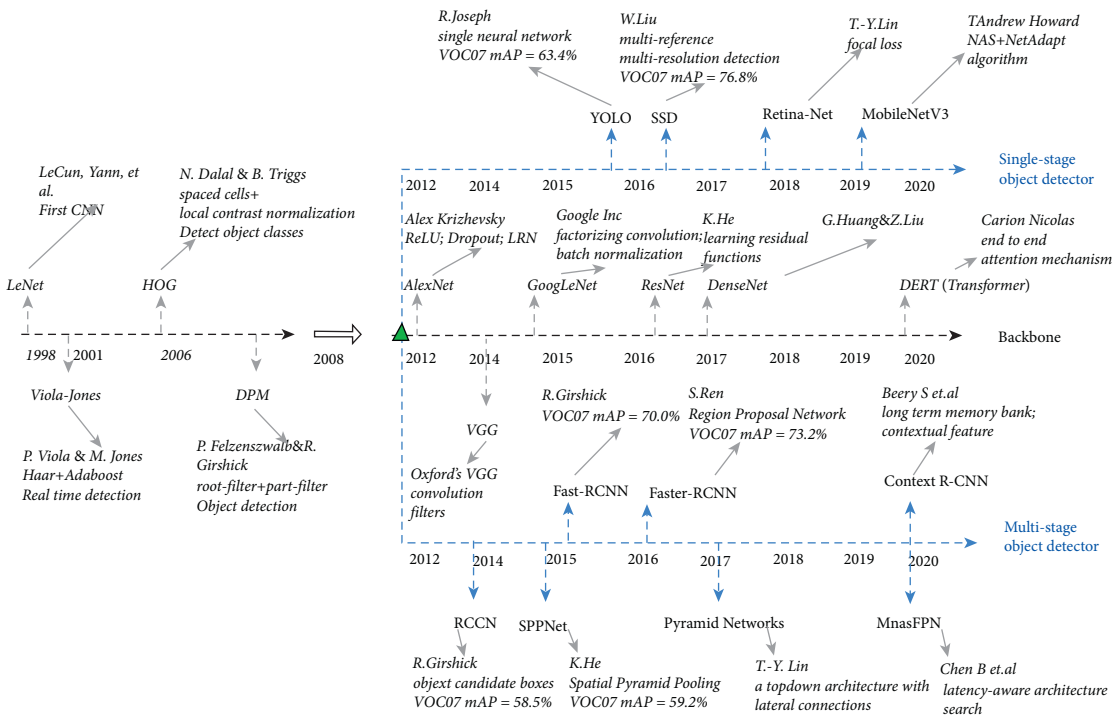


Figure 2: Deep learning-based visual perception development of neural network backbone and object detector.

CNN has the ability of representation learning, and it can perform shift-invariant classification of input information according to its hierarchical structure.

LeNet is one of the earliest CNN. Since 1988 [38], after many successful iterations, this pioneering result completed by Yann LeCun was named LeNet5. The architecture of LeNet5 is based on the view that the features of an image are distributed across the entire image, and the convolution of learnable parameters is an effective way to extract similar features in multiple locations with a small number of parameters.

In the nearly 20 years since LeNet was proposed, neural networks were once surpassed by other machine learning methods, such as support vector machines. Although LeNet can achieve good results on early small datasets, its performance on larger real datasets is not satisfactory. Computationally complex and insufficient computing power are the two main reasons for limiting its performance.

In 2012, Alex Krizhevsky proposed AlexNet [39]. Specifically, there are the following four innovations: (a) the GPU is used for network acceleration training for the first time. (b) ReLU activation function is used instead of the traditional sigmoid activation function and tanh activation function. (c) LRN local is used for response normalization. (d) In the first two layers of the fully connected layer, the dropout method is used to randomly inactivate neurons in a certain proportion to reduce overfitting.

AlexNet adds 3 convolutional layers on the basis of LeNet. VGG [40] proposed the idea of building a deep model by reusing simple basic blocks. The convolutional layers of vgg-block have the same structure, which means that the input size is equal to the output size. VGG proposed the idea of building a deep model by reusing a basic vgg-block. All VGG-block configurations are designed using the same principles, the filter (kernel) adapted with a very small receptive field (3×3), the convolution stride is fixed to 1 pixel, the padding is used to maintain the image resolution after convolution, and the max-pooling is performed over a 2×2 pixel window, with stride 2. This design increased network depth to improve classification accuracy.

In 2014, GoogLeNet [41] proposed the inception network structure, which is to construct a "basic neuron" structure to build a network structure with sparseness and high computational performance. The following two innovations should be highlighted: (a) using factorization into small convolution can reduce the number of parameters, reduce overfitting, and increase the nonlinear expression ability; (b) using the Inception Module, multiple branches extract high-level features with different levels of abstraction, which can enrich the expressive ability.

In some practice, the training error tends to increase instead of decreasing after adding too many layers. Even if the numerical stability brought by batch normalization makes it easier to train deep models, the problem still exists. He et al. [42, 43] presented a residual block (ResNet) to solve this problem. The ResNet can train an effective deep neural network through the cross-layer data channel; it deeply influenced the design of later deep neural networks [44–47].

The cross-layer connection design in ResNet has led to several follow-up works, and DenseNet [48] is one of the representative innovations. The main building blocks of DenseNet are dense block and transition layer. The former defines how the input and output are connected, and the latter is used to control the number of channels so that it is not too large.

In the field of computer vision, CNN has always occupied the mainstream position. However, researchers continue to try to introduce the transformer model in the field of natural

language processing (NLP) into computer vision, propose a new Vision Transformer model, and achieve performance close to the current SOTA method on multiple image process benchmarks. DERT [49] demonstrated that the transformer model for NLP can also be used for image pretraining and object detection tasks. Han et al. [50] surveyed the research of transformer-based computer vision.

Deep learning-based object detection models still have to solve the three problems of region selection, feature extraction, and classification regression. Generally speaking, it can be divided into two categories: single-stage methods and multistage methods.

The multistage methods have high localization and object recognition accuracy, and the example models include R-CNN [51], SPPNet [52], fast R-CNN [53], faster R-CNN [54], mask R-CNN [55], and cascade R-CNN [56]. The R-CNN framework is a typical representative of the multistage method. It uses selective search to generate candidate regions and then the detection process, and the number of candidate windows is controlled at about 2000. After selecting these image frames, the corresponding frames can be resized and then sent to CNN for training. Due to the very powerful nonlinear characterization ability of CNN, it can perform good feature expressions for each region. The final output of CNN uses multiple classifiers for classification judgment. This method increases the detection rate on PASCAL VOC [57] from 35.1% to 53.7%, which is equivalent to AlexNet's breakthrough in classification tasks in 2012 and has a profound impact on the field of target detection. Subsequently, Fast R-CNN proposed RoI Pooling to select regional features from the convolutional feature map corresponding to the entire image, which solved the problem of repeated feature extraction. Faster R-CNN proposes region proposal, anchors divide the image into n^*n regions, and each region gives 9 proposals with different ratios and scales, which solves the problem of repeatedly extracting candidate proposals. Other representative multistage object detectors also include SPPNet [52], pyramid networks [58], context R-CNN [59], and MnasFPN [60].

Single-stage methods prioritize inference speed, and example models include YOLO [61], SSD [32], RetinaNet [62], and MobileNetV3 [63]. YOLO is the representative single-stage model; there is no explicit bounding box extraction process. First, it resizes the image with a fixed size, divides the input image as a 7×7 grid, predicts 2 bounding boxes per grid, and classifies and locates for each bounding box. The YOLO model has also undergone many versions of development and is currently developed to YOLOv5. YOLO's approach is fast, but there will be many missing objects, especially tiny objects. So, single shot multibox detector (SSD) adds the concept of anchor from Faster R-CNN on the basis of YOLO and combines the features of different convolutional layers to make predictions. The main contribution of SSD is the multireference and multiresolution detection techniques, which significantly improve the detection accuracy of a one-stage detector, especially for some tiny objects [64]. Although the methods of the YOLO and SSD series do not have the extraction of region proposals and it becomes faster, they inevitably lose information and accuracy. The more representative single-stage object detector also includes RetinaNet [62] and MobileNet [63] series models.

In the field of computer vision, commonly used datasets include Microsoft Common Objects in Context (MSCOCO) [65], Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) [66], Visual Genome [67], Dataset for Object deTection in Aerial Images (DOTA) [68], and PASCAL Visual Object Classes Challenge (PASCAL VOC) [57]. The most popular object detection benchmark is the MSCOCO dataset. Models are typically evaluated according to a mean average precision metric.

In the field of marine object detection, there are few specialized datasets related to maritime supervision and autonomous ship navigation. Zhang et al. [69] proposed the use of generative adversarial networks (GANs) to solve insufficient marine data when training some object detection neural network. In [70], the novel idea is extracting the mask of the foreground object and combining it with the new background to automatically generate the location information and object information. Marine object detection-related dataset will be introduced in the subsequent section, separately.

2.3. Optimization Methods. The marine environment is complex and changeable, and the visual data has its own characteristics. Therefore, most of the researchers optimize the model and enhance the data based on the characteristics of the marine environment, to improve the accuracy and speed of marine object detection.

Chen et al. [71] presented a novel hybrid deep learning algorithm that combines improved generative adversarial network (GAN) and CNN-based detection methods for small ship detection. It uses Gaussian Mixture Wasserstein GAN with gradient penalty to generate sufficient informative artificial samples of small ships and uses raw and generated data to approach high accuracy tiny object detection. Ren et al. [72] proposed an effective ship image recognition method, which combines Hu invariant moment features and CNN features to achieve superior ship image recognition. Hu moment invariant feature joint to the last pooling layer achieves the highest recognition accuracy on self-built and VAIS datasets. Cao et al. [73] proposed a ship recognition method based on Morphological Watershed image segmentation and Zemike moment; although the Hu moment and Zernike moment are geometrically invariant, the Hu moment is unstable when the scale changes and the Zemike moment has a better stability. Using rotation to enhance the dataset causes errors in object detection tasks, Dong et al. [74] proposed a multiangle box-based rotation insensitive object detection structure (MRI-CNN) that improves the robustness of the model and reduces the detection performance impact due to the insufficient dataset.

3. Marine Target Detection Application

3.1. Maritime Surveillance. The interest in maritime surveillance has been increased in the last decades, and it is a significant issue for assuring the safety and security of

international transportation and defense mission. Despite being an important activity, how to efficiently conduct maritime surveillance is still a difficult problem for all countries. Computer vision-based digital maritime surveillance can solve most of this situation awareness issues, which can be divided into three categories: (1) detection and location (e.g., manmade pollution, oil spills, maritime hazardous event, noxious substances, and crashed plane debris), (2) tracking (e.g., ships, shipwrecks, lifeboats, illegal fisheries, illegal ballast water discharge, and smuggling), and (3) behavioral recognition (e.g., abnormal path confirmation, ships rendezvous, and high-speed objects on maritime surface). Most researchers focused on shore-based maritime surveillance, high-resolution satellite image surveillance, synthetic aperture radar (SAR) remote sensing, and so on.

3.1.1. Shore-Based Surveillance. In the current social environment, the traditional marine video surveillance technology simply relies on a large number of maritime managers who are no longer able to meet the needs of safe navigation. Computer vision combined with image processing technology has become the mainstream of maritime surveillance. In [75], experiments show that the ship detection based on YOLOv3 has high accuracy in the face of different scenes such as small traffic flow, foggy ship navigation, large traffic flow, and small imaging scale. The YOLOv3 algorithm uses the *k*-means algorithm to predict the bounding box and combines the multiscale features for ship identification; YOLOv3 can adapt to port scenarios with different traffic flow by using a multiscale detection mechanism, which has strong generalization ability.

Shao et al. [76] proposed a ship detection model based on a saliency-aware CNN framework that realizes real-time detection through the monitoring video taken by the camera. It can predict the category and position of the ship and use the global contrast-based salient region detection to correct the location. Based on the YOLOv2 pipeline, a saliency-aware CNN framework is proposed to improve the accuracy and robustness of ship detection under complex coastal conditions. Liu et al. [77] improved the YOLOv3 anchor method and feature fusion structure, respectively, GIOU loss was added to the loss function, and cross PANet was proposed to replace the FPN structure in YOLOv3. The results show that the proposed method can significantly improve the accuracy of YOLOv3 detecting sea surface objects. The SeaBuoys dataset was established according to the actual sea surface conditions, and comparative experiments were carried out with the existing SeaShips dataset [78].

Li et al. [79] proposed a new ship detection from visual image (SDVI) algorithm, named enhanced YOLOv3 tiny network for real-time ship detection. The convolution layer, instead of the max-pooling layer and expanding the channels of the prediction network, introduced an attention module named CBAM into the backbone network, which makes the model more focus on the target. The algorithm has a 9.6% improvement in mAP and has a faster detection speed. Huang et al. [80] used *k*-means++ clustering on the dimensions of bounding boxes to prioritize the model, improve the YOLOv3-Darnet53 network, increase jump connection mechanism, decrease feature redundancy, and improve the ability of tiny ship detection. On the premise of ensuring real-time performance, the precision of ship identification is improved by 12.5%, and the recall rate is increased by 11.5%.

In [12], a "reference model" pretrained with Pascal VOC image dataset and a "proposed model" trained with a specific maritime dataset (Singapore Maritime Dataset, SMD), the same structure of the "reference model" compared with the "proposed model," experiments show that, in SMD verification dataset, the proposed model is about twice as accurate as the reference model in terms of IoU and recall rate. Cane et al. [81] evaluated semantic segmentation networks in the context of an object detection system for maritime surveillance. The authors indicate that the SegNet and ENet achieve higher detection accuracy and precision. Considering the maritime surveillance actual condition, the ENet model would be the most suitable model.

3.1.2. High-Resolution Satellite Image Surveillance. High-resolution color remote sensing ship images taken from short distances provide advantages in ship detection applications. But the analysis of these high-dimensional images is complicated and requires a long time [36]. Synthetic aperture radar (SAR) is an active side-looking radar that can overcome weather interference and provide high-resolution images. SAR creates two-dimensional images or three-dimensional reconstructions of objects; it is typically mounted on a moving platform, such as an aircraft or spacecraft, and has its origins in an advanced form of side-looking airborne radar (SLAR).

Ghosh [82] proposed an efficient onboard detection system connected with a medium resolution wide amplitude optical camera and solved the problem of limited satellite coverage and limited simulation and equipment. Tian et al. [83] proposed a detection framework based on remote sensing image combining image enhancement module and dense feature reuse module to improve the object detection capability. Chen et al. [84] proposed an improved YOLOv3 based on an attention mechanism for fast and accurate ship detection, which accelerates detection speed to achieve real-time detection effect and improves the level of maritime surveillance.

Wang et al. [85] proposed an improved YOLOv3 algorithm for ship detection in optical remote sensing images. Adding the squeeze-and-excitation (SE) structure to the backbone improves the feature extraction capabilities and improves the detection accuracy by the fusion of multiscale feature maps. It achieves detection speeds of about 27 fps on NVIDIA RTX2080ti, with recall (R) = 95.32% and precision (P) = 95.62%. Cao et al. [86] conducted a similar study, the feature pyramid structure is introduced to combine the deep semantic information with the shallow semantic information, and the multiscale feature mapping is integrated to improve the detection ability of small objects.

Tang et al. [87] proposed a ship detection method based on noise classification and target extraction. The method consists of three modules: NLC (noising level classifying) module, STPAE (SAR target potential area extraction module) module, and the recognition module based on YOLOv5. The advantage of this model is that it can reduce the noise interference from the coast to ship detection. Tang et al. [88] introduced a novel high-resolution image network-based approach based on the preselection of a region of interest (RoI). It designs an HSV (hue, saturation, and value) module composed of four cores: background removal, noise removal, box-finding, and noise deletion, which can obtain useful RoI in a short time.

3.1.3. Airborne Maritime Surveillance. The speed of maritime surveillance ships is difficult to operate in complex seas and/or dispatching in busy ports. At present, although some maritime regulatory agencies carry manned helicopters, manned helicopters cannot take off to ensure the safety of personnel under hazardous sea conditions. At the same time, the cost of use is high; it is unable to meet the high-density, high-intensity maritime surveillance requirements.

Unmanned aerial vehicles (UAVs) can be remotely controlled or fly in the air in autonomous mode. It is a miniaturized and intelligent flight platform that can complete one or more tasks by carrying different task modules. It also has great potential in maritime applications. Solving the problems of drone flight stability, data transmission, shipboard electromagnetic compatibility, and convenient take-off and landing in the marine environment can enable drones to play a greater role in maritime applications. At present, according to the principle of flight, UAVs suitable for maritime applications mainly include fixed-wing UAVs, unmanned helicopters, multirotor UAVs, and vertical take-off and landing fixed-wing UAVs. Various types of UAVs have their own unique advantages in maritime applications.

Ribeiro et al. [89] presented an airborne maritime surveillance dataset captured by a small size UAV. This dataset presents object examples ranging from cargo ships, small boats, and life rafts to oil spill. Due to the continuous shaking of the UAV's camera, it is very difficult to label data on the acquired video dataset. The authors proposed a new labeling tool, which is developed in C++, and the OpenCV library is used to create labels manually. Reference [90] presents an approach to detect boats in a maritime surveillance scenario using a small UAV. This work relies on CNN to perform robust detection even in the presence of distractors like wave crests and sun glare. Reference [91] explores maritime search and rescue missions by using experimental UAV data to detect the sea surface object. Reference [92] addresses the development of an integrated system to support maritime situation awareness based on UAVs, emphasizing the role of the automatic detection subsystem.

Xiu et al. [93] contributed a system that includes a maritime unmanned aerial vehicle (Mar-UAV) with a high-resolution camera and an Automatic Identification System (AIS). Multifeature information, including position, scale, heading, and speed, is used to match between real-time image and AIS message. The results demonstrate that the proposed algorithm and the Mar-UAV system are very significant for achieving autonomous maritime surveillance. Reference [94] presents a method to learn spatial and temporal features from video sequences; temporal features attempt to improve the maritime objects detection ability, which contain strong distractors such as glare and wakes. The proposed method is composed of two main parts, one spatial feature extractor based on the VGG network and one recurrent layer, the ConvLSTM.

3.1.4. Satellite Radar Image Surveillance. Ship detection in synthetic aperture radar (SAR) images has been widely studied due to its indispensable role in military intelligence acquisition, maritime management, and many civil fields. However, due to the limitations of bandwidth and computer computing power in satellite scenarios, SAR image-based ship detection deployment is largely hindered. Another reason is that searching for targets of interest in massive SAR images by eyes becomes time-consuming and often impractical. Therefore, lightweight neural network training models are widely used.

Chen et al. [95] proposed a novel learning scheme for training a lightweight ship detector called Tiny YOLO-Lite, which simultaneously (1) reduces the model storage size; (2) decreases the floating-point operations (FLOPs) calculation; and (3) guarantees the high accuracy with faster speed. Reference [96] proposes a lightweight CNN-LiraNet combining dense connections, residual connections, and group convolution. It uses a two-layer predictor and adds residual models to transmit features easier; experimental results show that the Lira-YOLO network has less complexity, only 2.980 Bflops. The parameters only have 4.3 MB. The mean average accuracy (mAP) index of the Mini-RD and SARShip detection dataset (SSDD) reaches 83.21% and 85.46%, respectively, which is comparable to the tiny-YOLOv3.

Yang et al. [97] proposed a one-stage object detection framework based on RetinaNet and rotatable bounding box (RBox) for the problems such as feature scale mismatch and task contradiction. Experimental results show that the average accuracy improved 13.26%, 9.29%, 8.92%, 8.55%, and 4.55% compared to the other four advanced RBox-based ship detection methods at the IoU threshold of 0.5. In this paper, scale calibration is proposed to make the proportion distribution of the main feature map and the object feature map consistent.

In arctic waters, a vast majority of objects are icebergs drifting in the ocean and can be mistaken for ships in terms of navigation and ocean surveillance. Hass and Jokar Arsanjani [98] presented a YOLOv3-based deep learning model that uses SAR images to discriminate icebergs and ships, which could be used for mapping ocean objects ahead of a journey.

To solve the problem of small objects and multiobject ship detection in complex scenarios, [99] proposes a detection method based on an optimized feature pyramid network (FPN) model. The results show that the small ship

detection accuracy reaches 98.62%, and the proposed model has higher accuracy and better comprehensive performance compared with YOLO.

3.1.5. Other Applications. For military defense and intelligent early warning, an infrared intrusion object detection algorithm based on a neural network is proposed [100]. The extended CNN designed by this algorithm can fuse and expand the image features, enhance object filtering, and improve background suppression. Xie et al. [101] proposed an inspection system based on tracking technology, which can automatically process ship inspection video and predict suspicious areas where cracks may exist. Intelligent computer vision is the most important technology for the development and utilization of deep-sea resources. Han et al. proposed a combination of the max-RGB method and shades of the gray method that is applied to achieve the enhancement of underwater vision [102]. In [103], vision-based object detection for underwater robots has been proposed. In order to overcome the limitations of cameras and to make use of the advantages of image data, a number of approaches have been tested. The topics include color restoration algorithm for the degraded underwater images, detection, and tracking methods for underwater target objects.

3.2. Vision-Based Autonomous Ship Navigation. Object detection and vision-based ship navigation is an essential task for autonomous ship navigation. However, sunlight reflection, camera motion, and illumination changes may cause false object detection in the maritime environment. Farahnakian and Heikkonen [104] proposed three fusion architectures (pixel-level, feature-level, and decision-level) to fuse two imaging modes (visible and infrared); they employed deep learning for performing fusion and detection. Pan et al. [105] proposed the navigation mark classification and identification model based on deep learning (RMA: ResNet-Multiscale-Attention), which can identify different navigation marks finely. It can identify the nuances of the navigation mark; no additional supervision information is required except for the label, and it is end-to-end training.

3.2.1. Horizon Detection. Marine horizon detection is the most significant semantic boundary for segmenting the image into sea and sky. References [18, 19] have summarized the marine horizon line detection. In the past research, many robust marine horizon generation methods have been proposed. For the marine horizon model of the straight line, the traditional methods include the following: (a) linear fitting: the selection of candidate points of this method is easily susceptible to the complex sea and sun glint [106]. (b) Image segmentation: the optimal segmentation threshold of this method is difficult to be adaptively determined [107]. The algorithm's anti-noise ability is insufficient. (c) Gradient significance, each interference factor has abundant edges, and these edges have gradient values similar to or even higher than the marine horizon, which is easy to cause false detection [108].

Typical marine horizon detection relies on edge information, which requires two important issues to be overcome: unstable edge detection and complex marine environment with shore background and weather conditions. Jeong et al. [109] proposed a novel method for horizon detection that combines a multiscale approach and CNN; it has a median positional error (MPE) of less than 1.7 pixels from the center of the horizon and a median angular error (MAE) of approximately 0.1 degrees. This method is one of the methods for horizon detection with high speed and high accuracy, but it may have failed detection in some scenarios such as an absence of obvious line feature. Prasad et al. [110] presented a novel method called multiscale consistence of weighted edge Radon transform, abbreviated as MuSCoWERT. It has a median error of about 2 pixels (less than 0.2%) from the center of the actual horizon and a median angular error of less than 0.4 deg. Compared with traditional methods (ENIW [111], FGSL [112], MuSMF [113], IntGF, IntG, Hough [114], and GWR [115]), MuSCoWERT has excellent performance. Jeong et al. [116] proposed a fast method for detecting the horizon line in maritime scenarios by combining a multiscale approach and region-of-interest detection. Experimental results show that the proposed method can accurately identify the region of interest on the moving platform and ensure the robustness of sea-sky-line detection. And it is less affected by ships, light changes, waves, and wakes. In [117], a novel algorithm based on probability distribution and physical characteristics is introduced. The authors designed a hybrid method, which consists of sea-sky region extraction and horizon estimation based on the information of color, texture, and context. The proposed algorithm precisely detects the horizon not only from fine images but also from blurred image, even with a splashed camera.

As shown in Table 1, mean height deviation (MHD) and angle deviation (AD) as the evaluation standard have been recognized by most of the researchers. Nondeep learning methods occupy the majority of marine horizon detection. In recent years, marine horizon detection based on the Singapore Maritime Dataset has gradually increased, which is conducive to the comparison between different algorithms. Although it is still affected by the objective environment such as the different computing power of computers, it will play a role in pointing the direction of future research.

Accurate identification, tracking, and positioning of the marine horizon, as well as an accurate description of water boundary lines, are the basic requirements for safe driving of autonomous ship navigation. However, a large number of current researches mainly focus on pure marine horizon detection, and there is no in-depth research on marine horizon tracking and positioning and accurate description.

From the perspective of the marine horizon detection process, this survey summarizes and analyzes the key points of the existing marine horizon detection methods and summarizes the content that still needs to be studied in the future. It is suggested that, in complex water environment

Table 1: Examples of typical marine horizon detection algorithms.

Methods	Dataset	Advantage	Disadvantage
Zhang et al. [118]	MAV	Correct ratio >99.9%.	—
Lipschutz et al. [119], probability distribution edge detection Hough transform	—	Visible-light images: MHD ~3.69 pixels; AD ~0.28 degrees. Infrared images: MHD ~1.49 pixels; AD ~0.14 degrees	—
Gershikov et al. [114], H-REM	—	MHD ~2.28 pixels; AD ~0.19 degree; mean run time ~0.14 s	—
Prasad et al. [110], weighted edge Radon multiscale consistence	SMD MAR-DCT Buoy dataset	MPE ~2 pixels; MAE ~0.4 degrees	Does not work well in certain scenarios
Jeong et al. [116], multiscale approach region of interest (RoI)	SMD Buoy dataset	15 fps; MPE <2 pixels; MAE ~0.15 degrees	Performance reduction (edges related to horizon)
Sun et al. [120], coarse-fine-stitched hybrid filtering Random sample	SMD Marine Obstacle	MHD ~0.89 pixels; MAD ~0.19 degrees	—
Liang et al. [117], probability distribution physical characteristics	SMD Buoy dataset	MPE ~7.6 pixels; MAE ~0.4 degrees	Ineffective (large area occlusion)
Jeong et al. [109], multiscale approach NN	SMD	MPE <1.7 pixels; MAE ~0.1 degrees	Line features absent
Yang et al. [121], probabilistic graphical expectation-max Gaussian models	Marine obstacle	—	Reflection and illumination

and engineering applications with high real-time requirements, marine horizon detection is facing severe challenges. In future work, we should improve the algorithm to improve the real-time performance and environmental adaptability of the algorithm.

3.2.2. Surface Moving Object Detection. Over the last decades, a lot of researchers have worked on the big challenge of detection of moving ships in various complex marine environments. Reference [122] presents a ship object detection algorithm to achieve efficient visual maritime surveillance from nonstationary surface platforms.

The maritime target detection represented by YOLO has made great achievements in recent years. Chen et al. [123] proposed a YOLO-based integrated framework to detect ships from maritime surveillance videos and accurately identify ship behavior in continuous frames. The average check rate reaches 92.85%, and the registration rate reaches 93.91%, respectively. It shows that the proposed method identifies the historical behavior of the detected object successfully, helps managers understand the historical navigation, predicts the future navigation trajectory, implements early warning measures to ensure maritime traffic safety. Li et al. [124] proposed a lightweight ship detection model (LSDM) based on YOLOv3 and DenseNet, in which the backbone network is improved by using dense connection inspired from DenseNet, and the feature pyramid networks are improved by using spatial separation convolution to replace the original convolution network. In the proposed model, only one-third of the parameters of the YOLOv3 network can reach average accuracy of 94% for ship detection, and in the LSDM tiny network, just one-eighth of the parameters of the YOLOv3 network can reach double detection speed and average accuracy of 93.5%.

Qiao et al. [125] proposed a detection framework based on YOLOv3, which integrates multimodel and multicue (M^3C) pipeline. Multimodel is used to solve the problem of unstable tracking of target maneuverability in traditional single-model Kalman tracker (such as CV model), and multicue solves the problem of frequent IDS caused by motion blurring and occlusion. The two public maritime datasets showed that the proposed method achieved state-of-the-art performance, not only in identity switches (IDS) but also in frame rates. Huang et al. [126] solved the problem of low recognition rate on a small dataset and improved the real-time performance of ship detection. It provides a high-precision, real-time ship detection for smart port management and USV visualization.

The author discovered that the current research of marine moving object detection has flaws, and the dataset from the perspective of the ship bridge is difficult to obtain. So far, there is no suitable benchmark. Figure 3 shows an example of an onboard visual navigation dataset from the author's lab.

Benchmark datasets containing various marine scenes from the perspective of ship bridges need to be presented, and all relevant studies should have unified standards and recognized evaluation mechanisms. Figure 4 shows an example result of onboard object detection. In the verification dataset, the missed detection rate of small targets should be included, which is essential for the autonomous navigation of large ships.

As shown in the upper left corner of Figure 4, the obstacle of the marine surface and navigation aid signs should be clearly identified, and the trained model needs to

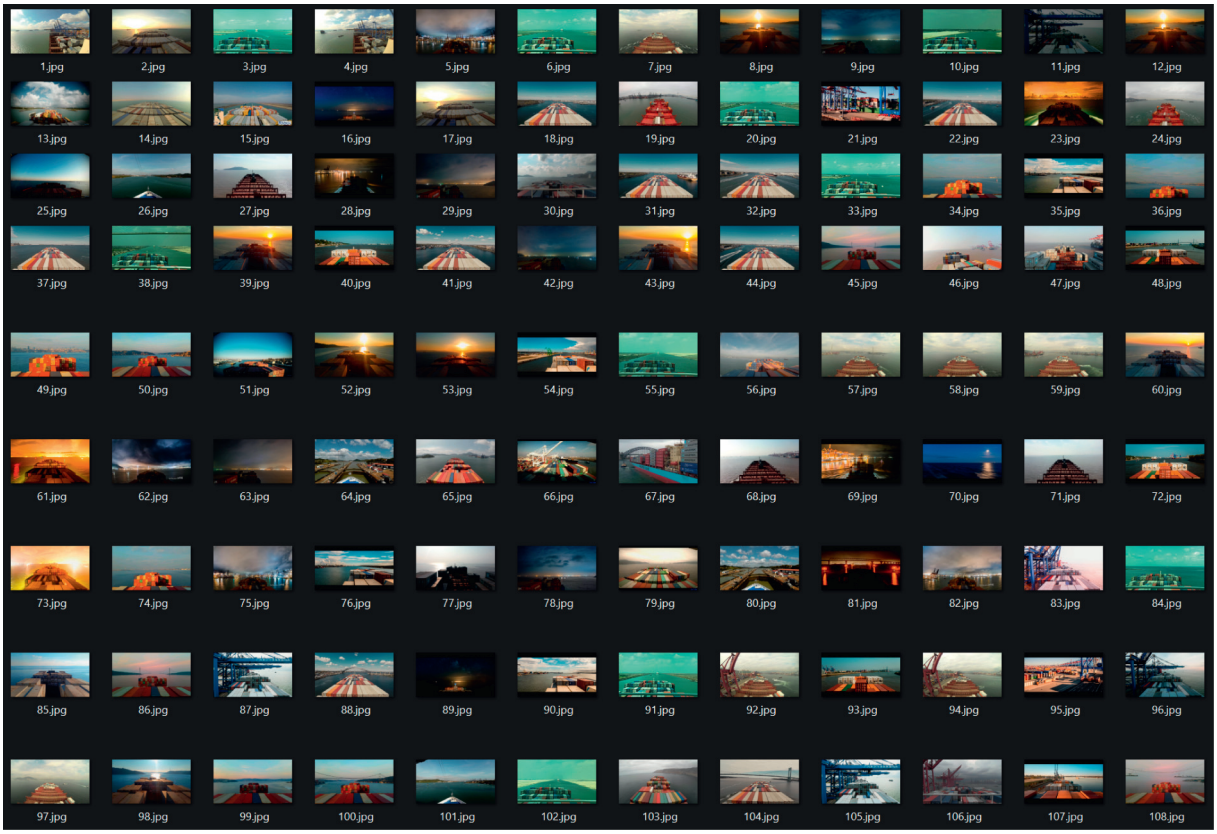


Figure 3: An example of onboard visual navigation dataset.

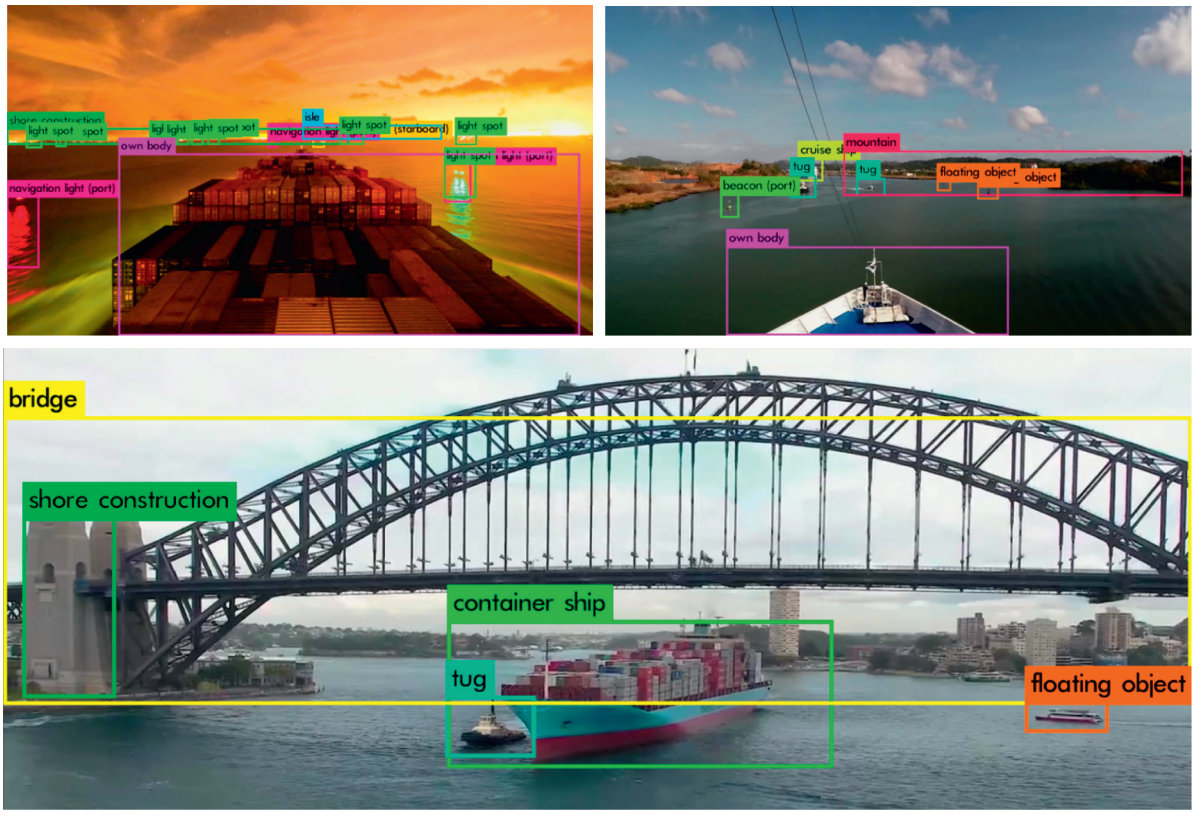


Figure 4: An example of onboard object detection.

understand the different meanings of different objects for navigation. The identification of near-shore constructions and moving objects on the water, the background lights on the shore, and the lights on the ship is still a critical problem for object detection.

3.2.3. Background Subtraction. In the characteristics of dynamic marine environment, the detector needs to subtract the dynamically changing objects from the backgrounds; meanwhile, there are a large number of linear features and constantly changing lighting conditions. Even the advanced sea level detection technology and video frame registration technology are facing challenges. Many background subtraction and object detection methods are very difficult in the video stream.

For example, [84] designs a multiclass ship dataset (MSD) to highlight the difference between the ship and the background; it can improve the accuracy of tiny ship detection.

Prasad et al. [127] provided a benchmark of the performance of 23 classical and state-of-the-art background subtraction algorithms on visible range and near-infrared range videos in the Singapore Maritime Dataset. This paper indicates the limitations of the conventional performance evaluation criteria for maritime vision and proposes new performance evaluation criteria that are better suited to this problem.

Although these 23 methods have been successful, the recall and accuracy are extremely low. Even the most advanced BS technology cannot deal well in the marine environment. This means that the new BS algorithm needs to be formulated for maritime vision. The traditional performance evaluation index IoU is modified to a new evaluation index IOG, and a new index bottom edge proximity (BEP) is proposed to judge whether the bottom of detection object (DO) and ground truth (GT) are close. This indicator enables more extensive detection in the presence of trails.

Zhang et al. [122] proposed a discrete cosine transform (DCT) based ship detection algorithm which can extract the sea regions accurately for complex background modeling. The main contribution is to provide more accurate detection results within the complex sea surface background, which is of vital importance for ship-/buoy-based surveillance applications in the presence of large waves. The independent detectors for sky and sea regions increase the detection sensitivity to small objects around the horizon.

The lighting environment at sea is ever-changing; one method or a model suitable for one weather and lighting condition is ineffective. Establishing a model that can seamlessly select models and methods for different lighting conditions is essential for the practical application of maritime treatment. Prasad et al. [13] discussed the technical challenges in maritime image processing and machine vision problems for video streams generated by cameras. Challenges are arising from the dynamic nature of the background, unavailability of static cues, presence of small objects at distant backgrounds, and illumination effects.

Chan et al. [128] compared thirty-seven nonstatic electrooptical sensor (combine visible-light and infrared cameras)-based background subtraction methods; the results indicate that background subtraction algorithms of the multiple features category can better handle maritime challenges, thereby realizing higher accuracy when analyzing visible-light and infrared cameras.

3.2.4. Other Applications. Augmented reality (AR) can combine computer-generated graphic information with real camera views and is an effective display technology. Reference [129] used additional location data retrieved from the AIS device to improve retrieval performance based on the characteristics of the sea-sky-line boundary and used the *k*-means clustering algorithm and pixel contour to distinguish the sea-sky-line. The author also emphasized that the proposed system is based on CCTV and computer image processing; therefore, the performance is influenced by sea conditions, for example, the low light condition such as foggy, dark-night, and heavy rainy days.

4. Discussion

4.1. Model Comparison. The accuracy and real-time requirements of object detection for autonomous ship navigation and maritime surveillance are important. It is necessary to propose a maritime environment image/video perception based on an improved regressive deep convolution network. YOLO series architecture is always the first neural network to be considered, for example, [12, 71, 74, 75, 77, 79, 82, 84, 88, 123, 126, 130–132]; these improvements contributed to a stronger baseline cross YOLO series detector.

As shown in Table 2, we collect some YOLO backbone network-based marine object detection models. Based on the advantages of YOLO in detection efficiency and speed, most of the researches focus on ship detection task. Experiments on public datasets (such as SMD and SeaShips) show that most of the enhanced YOLO series models have improved performance in different levels.

4.2. Marine Datasets Comparison. Moosbauer et al. [144] proposed a benchmark that is based on the Singapore Maritime Dataset (SMD). As shown in Table 3, this dataset included onshore and onboard objects in the marine environment; it provides Visual-Optical and Near-Infrared videos along with annotations for object detection. The authors evaluate two state-of-the-art object detection models for the applicability in the maritime domain: Faster R-CNN and Mask R-CNN. The SMD-based dataset can be used as a benchmark that encourages reproducibility and comparability for object detection in maritime environments. Recent research [12, 70, 81, 110, 127, 144–151] reflects this characteristic.

To advance object detection research in Earth Vision, also known as Earth Observation and Remote Sensing, [68] introduces a large-scale Dataset for Object deTection in Aerial images (DOTA). There are many studies using this

Table 2: Application of improved YOLO backbone network in maritime object detection.

Algorithms (backbone)	Datasets	Scenarios	Improved method	Effect
YOLOv2 [71]	Small ship dataset	Small ship detection	Density-based spatial clustering (DBSCAN)	AUC: 0.960 TPR: 98.3% FPR: 3.5%
YOLOv3 [77]	SeaShip dataset Buoy dataset	Ship detection	Loss function (GIOU) PANet replaces FPN	mAP (SeaShip): 98.37% mAP (Buoy dataset): 90.58%
YOLOv2 and CNN [12]	Pascal VOC SMD	Ship detection	—	Recall: 77.12% IoU: 66.69%
YOLOv3 [75]	Shanghai port surveillance video	Ship detection	—	Average acc.: 0.84
YOLOv3 [79]	SeaShip dataset	Ship detection	CBAM	mAP increase 9.6%
YOLO [123]	Self-collected	Ship detection	—	Average acc.: 92.85%
YOLOv3 tiny [124]	From Internet	Ship detection	Dense connection spatial separate conv.	LSDM average acc.: 94% LSDM tiny: 93.5%
YOLOv3 [132]	LWIR	Object detection	—	mAP@0.25 IoU: 0.97 mAP@0.50 IoU: 0.90 mAP@0.75 IoU: 0.29
YOLOv3 [125]	SMD; PETS 2016	Ship tracking	—	mAP: 41.2%
YOLOv2 [130]	Pascal VOC SMD	Object detection Ship detection	Pass through layer Transfer learning	Recall: 73.86% IoU: 60.79%

Table 3: Comparison of representative marine object detection datasets.

Datasets	Resolution (Pixels)	Usage Scenarios	References	Data sources
SMD	1080 × 1920	Autonomous ship navigation Marine horizon detection Maritime surveillance Object detection Object tracking and so on	[13, 16, 18, 70, 109, 110, 133, 134]	Onboard and onshore
DOTA	4000 × 4000	Object detection Remote sensing Maritime safety Maritime surveillance	[135–138]	Aerial images
SeaShips	1920 × 1080	Marine object detection Target tracking Maritime surveillance	[78, 139, 140]	Onshore
AIR-SARShip	3000 × 3000	Object detection Remote sensing Maritime surveillance	[141–143]	GF-3 satellite

dataset in the field of maritime remote surveillance [135, 138, 152] and so on.

SeaShips is a large ship dataset. The dataset consists of 11,126 images, covering 6 common ship categories (ore ships, bulk carriers, general cargo ships, container ships, fishing ships, and passenger ships). All images come from about 5400 real video clips, collected by 156 surveillance cameras in the coastline video surveillance system. Some research uses this dataset to train their model or improve the model's performance [139, 140].

Some other datasets need to be highlighted. Spagnolo et al. [153] presented a boat Re-ID dataset composed of 107 classes, and each class represents a different boat with a total of 5523 images. In order to verify the superiority of the proposed dataset, the authors give the results of training CNN by using this dataset, and the research results can be used as a benchmark for future comparisons.

Bovcon et al. [154] introduced the MaSTr1325 dataset for training deep USV obstacle detection models in small-sized coastal USV. They also proposed a data augmentation protocol to address slight appearance differences. The dataset is applied to three popular semantic segmentation architectures: U-Net, PSPNet, and Deeplabv2, among which Deeplabv2 performs best in obstacle detection. In [148], the authors used 4K videos for maritime video surveillance and proposed an approach that attempts to leverage both temporal and spatial video information for achieving fast and accurate object extraction. Multiscale texture discrimination algorithm carried out key video locations to achieve final object extraction.

4.3. Current Challenges and Future Works. Computer vision is the subject of studying image information organization, object and scenario recognition, and interpreting events by taking images (video) as input and aiming at representation and understanding of the environment. Judging from the current research status, the research mainly focuses on the organization and recognition of image information, and the interpretation of events is rarely involved, at least at a very preliminary stage.

The relationship between artificial intelligence and computer vision is as follows: artificial intelligence puts more emphasis on reasoning and decision-making, but at least computer vision is still mainly at the stage of image information expression and object recognition. Object recognition, environment perception, and scenario understanding also involve reasoning and decision-making from image features, but they are fundamentally different from the reasoning and decision-making of artificial intelligence.

4.3.1. Current Challenges. Specific maritime engineering applications belong to systemic issues, affected by many objective factors, for example, equipment shaking, model dependence, and light interference on shore.

(i) Shaking problem of imaging equipment: in actual marine engineering applications (onshore and onboard), the effect of the model is often much lower than the accuracy and speed obtained in the laboratory. In the actual marine environment, the shaking of the equipment is the main reason for small object missed or object false detection. Even in the tracking task, the tracking fails due to the same reason.

(ii) Model dependence: at present, all models require fixed scene training; the environmental changes have a large impact on the recognition accuracy of the model. Weather changes will change the external photosensitive environment, which will lead to bad results for marine object detection, and even equipment updates can cause model detection to fail.

(iii) Background light pollution on the shore is an important issue, and there are few research papers related to the extraction of background light on the shore. Even experienced seafarers are still prone to think of shore lights as lights moving on the sea and make inappropriate decision-making. This is an urgent problem in the field of autonomous ship navigation.

4.3.2. Future Works

(1) Online Training. The current models are trained first and then deployed. Applications such as autonomous ship navigation require reasoning and decision-making based on environmental information in real time. One of the future trends is to solve this problem.

(2) Build a Maritime Data Sharing Center. (1) Unified model algorithm evaluation mechanism: at present, maritime surveillance and intelligent transportation need a public benchmark for different researchers who proposed various models. (2) Construct various marine scenarios and sea condition dataset share platform. The actual detection task of marine objects requires training with a lot of data in their respective sea conditions; the current datasets cannot complete this task. We will put more energy into the work of data sorting and build a maritime data sharing platform.

5. Conclusions

This survey covers most of the application scenarios of object detection for maritime surveillance and autonomous ship navigation. In recent years, a large number of marine object detection models based on deep learning have been proposed, but due to the lack of universal evaluation criteria, it is difficult to compare different improved models. According to the characteristics of the maritime environment, this paper summarized the advantages of the computer vision milestone model and proposed different application scenarios of the single-stage model and the multistage model under different development routes. The most popular YOLO series models are compared in different dimensions, and the importance of public dataset benchmarks is proposed. We also discussed the urgency of building a maritime proprietary dataset platform that satisfies different scenarios and model training in practical engineering applications. This work will put forward feasible suggestions for future research directions of deep learning-based marine object detection.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities, Grant nos. 3132021130 and 3132019400.

References

[1] A. Felski and K. Zwolak, "The ocean-going autonomous ship-challenges and threats," *Journal of Marine Science and Engineering*, vol. 8, no. 1, p. 41, 2020.

[2] A. Noel, K. Shreyanka, and K. Gowtham, "Autonomous ship navigation methods: a review," in *Proceedings of the Conference Proceedings of ICMET OMAN. 2019*, Muscat, Oman, November 2019.

[3] M. Furusho, "Visual environment and sight-line displacements of navigation officers for good lookout," *Journal of Light and Visual Environment*, vol. 25, no. 1, pp. 43–48, 2001.

[4] X.-Y. Zhou, J.-J. Huang, and F.-W. Wang, "A study of the application barriers to the use of autonomous ships posed by the good seamanship requirement of COLREGs," *Journal of Navigation*, vol. 73, no. 3, pp. 710–725, 2020.

[5] T. Morris, *Computer Vision and Image processing*, Palgrave Macmillan Ltd, London, UK, 2004.

[6] T. Cornsweet, *Visual perception*, Academic Press, Cambridge, MA, USA, 2012.

[7] Z. Zou, Z. Shi, and Y. Guo, "Object detection in 20 years: a survey," 2019, https://arxiv.org/abs/1905.05055.

[8] L. Jiao, F. Zhang, and F. Liu, "A survey of deep learning based object detection," *IEEE access*, vol. 7, pp. 128837–128868, 2019.

[9] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[10] J. Chen, "Foreground-background imbalance problem in deep object detectors: a review," in *Proceedings of the 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, Tokyo, Japan, August 2020.

[11] D. Qiao, G. Liu, and T. Lv, "Marine vision-based situational awareness using discriminative deep learning: a survey," *Journal of Marine Science and Engineering*, vol. 9, no. 4, p. 397, 2021.

[12] S. J. Lee, M. I. Roh, and H. W. Lee, "Image-based ship detection and classification for unmanned surface vehicle using real-time object detection neural networks," in *Proceedings of the 28th International Ocean and Polar Engineering Conference*, OnePetro, Sapporo, Japan, June 2018.

[13] D. K. Prasad, C. K. Prasath, and D. Rajan, "Challenges in video based object detection in maritime scenario using computer vision," 2016, https://arxiv.org/abs/1608.01079.

[14] S. Thombre, Z. Zhao, and H. Ramm-Schmidt, "Sensors and ai techniques for situational awareness in autonomous ships: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, 2020.

[15] M. N. Chapel and T. Bouwmans, "Moving objects detection with a moving camera: a comprehensive review," *Computer Science Review*, vol. 38, Article ID 100310, 2020.

[16] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[17] M. Moniruzzaman, S. M. S. Islam, and M. Bennamoun, "Deep learning on underwater marine object detection: a survey," in *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 150–160, Springer, Antwerp, Belgium, September 2017.

[18] M. A. Hashmani, M. Umair, and S. S. H. Rizvi, "A survey on edge detection based recent marine horizon line detection methods and their applications," *IEEE*, in *Proceedings of the 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–5, Sindh, Pakistan, January 2020.

[19] M. Petković, I. Vujović, and I. Kuzmanić, "An overview on horizon detection methods in maritime video surveillance," *Transactions on Maritime Science*, vol. 9, no. 1, pp. 106–112, 2020.

[20] B. Li, X. Xie, and X. Wei, "Ship detection and classification from optical remote sensing images: a survey," *Chinese Journal of Aeronautics*, vol. 34, 2020.

[21] Z. Liu, C. K. Loo, and K. Pasupa, "Meta-cognitive recurrent kernel online sequential extreme learning machine with kernel adaptive filter for concept drift handling," *Engineering Applications of Artificial Intelligence*, vol. 88, Article ID 103327, 2020.

[22] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*, vol. 1, Kauai, HI, USA, December 2001.

[23] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[24] G. Rätsch, T. Onoda, and K. R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[25] B. Yang, J. Yan, and Z. Lei, "Aggregate channel features for multi-view face detection," in *Proceedings of the IEEE International Joint Conference on Biometrics*, pp. 1–8, IEEE, Seoul, Korea, August 2014.

[26] M. Cerf, J. Harel, and W. Einhäuser, "Predicting human gaze using low-level saliency combined with face detection[J]," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1–7, 2008.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[28] Z. R. Wang, Y. L. Jia, and H. Huang, "Pedestrian detection using boosted hog features," in *Proceedings of the 2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 1155–1160, IEEE, Beijing, China, October 2008.

[29] G. Gan and J. Cheng, "Pedestrian detection based on HOG-LBP feature," in *Proceedings of the 2011 Seventh International Conference on Computational Intelligence and Security*, pp. 1184–1187, IEEE, Sanya, China, December 2011.

[30] Z. Luo, "Rotation-invariant histograms of oriented gradients for local patch robust representation," in *Proceedings of the 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, IEEE, Hong Kong, China, December 2015.

[31] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the 2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8, Anchorage, AK, USA, June 2008.

[32] W. Liu, D. Anguelov, and D. Erhan, "Ssd: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Amsterdam, The Netherlands, October 2016.

[33] C. Szegedy, W. Zaremba, and I. Sutskever, "Intriguing properties of neural networks," 2013, https://arxiv.org/abs/1312.6199.

[34] M. Oquab, L. Bottou, and I. Laptev, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717–1724, Columbus, OH, USA, June 2014.

[35] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proceedings of the European Conference on Computer Vision*, pp. 483–499, Springer, Zurich, Switzerland, September 2016.

[36] B. Dolapci and C. Özcan, "Automatic ship detection and classification using machine learning from remote sensing images on Apache Spark," *Journal of Intelligent Systems: Theory and Applications*, vol. 4, no. 2, pp. 94–102, 2021.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, vol. 1, MIT press, Cambridge, MA, USA, 2016.

[38] Y. LeCun, L. Bottou, and Y. Bengio, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[41] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[42] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[43] K. He, X. Zhang, and S. Ren, "Identity mappings in deep residual networks," in *Proceedings of the European Conference on Computer Vision*, pp. 630–645, Springer, Amsterdam, The Netherlands, October 2016.

[44] J. Y. Zhu, T. Park, and P. Isola, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, Venice, Italy, October 2017.

[45] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

[46] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI conference on artificial intelligence*, San Francisco, CA, USA, February 2017.

[47] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, December 2018.

[48] G. Huang, Z. Liu, and L. Van Der Maaten, "Densely Connected Convolutional networks," in *Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.

[49] N. Carion, F. Massa, and G. Synnaeve, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision*, pp. 213–229, Springer, Glasgow, UK, August 2020.

[50] K. Han, Y. Wang, and H. Chen, "A survey on visual transformer," 2020, https://arxiv.org/abs/2012.12556.

[51] R. Girshick, J. Donahue, and T. Darrell, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[52] K. He, X. Zhang, and S. Ren, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[53] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[54] S. Ren, K. He, and R. Girshick, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[55] K. He, G. Gkioxari, and P. Dollár, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, Venice, Italy, October 2017.

[56] Z. Cai and N. Vasconcelos, "Cascade R-Cnn: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, June 2018.

[57] M. Everingham, L. Van Gool, and C. K. I. Williams, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[58] T. Y. Lin, P. Dollár, and R. Girshick, "Feature Pyramid Networks for Object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[59] S. Beery, G. Wu, and V. Rathod, "Context r-cnn: long term temporal context for per-camera object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13075–13085, Seattle, WA, USA, June 2020.

[60] B. Chen, G. Ghiasi, and H. Liu, "Mnasfpn: learning latency-aware pyramid architecture for object detection on mobile devices," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13607–13616, Seattle, WA, USA, June 2020.

[61] J. Redmon, S. Divvala, and R. Girshick, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Seattle, WA, USA, June 2016.

[62] T. Y. Lin, P. Goyal, and R. Girshick, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, Venice, Italy, October 2017.

[63] A. Howard, M. Sandler, and G. Chu, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, Seoul, South Korea, October 2019.

[64] Á. Morera and A. B. Moreno, "SSD vs. Yolo for detection of outdoor urban advertising panels under multiple variabilities," *Sensors*, vol. 20, no. 16, p. 4587, 2020.

[65] T. Y. Lin, M. Maire, and S. Belongie, "Microsoft coco: Common objects in context," in *Proceedings of the European conference on computer vision*, pp. 740–755, Springer, Zurich, Switzerland, September 2014.

[66] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, Providence, RI, USA, June 2012.

[67] R. Krishna, Y. Zhu, and O. Groth, "Visual genome: connecting language and vision using crowdsourced dense image annotations," 2016, https://arxiv.org/abs/1602.07332.

[68] G. S. Xia, X. Bai, and J. Ding, "DOTA: a large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, Salt Lake City, UT, USA, June 2018.

[69] R. L. Zhang and M. Furusho, "Developing generative adversarial nets to extend training sets and optimize diiscrete actions," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 13, 2019.

[70] H. C. Shin, K. I. Lee, and C. E. Lee, "Data augmentation method of object detection for deep learning in maritime image," in *Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 463–466, IEEE, Busan, Korea, February 2020.

[71] Z. Chen, D. Chen, and Y. Zhang, "Deep learning for autonomous ship-oriented small ship detection," *Safety Science*, vol. 130, Article ID 104812, 2020.

[72] Y. Ren, J. Yang, and Q. Zhang, "Ship recognition based on Hu invariant moments and convolutional neural network for video surveillance," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1343–1373, 2021.

[73] X. Cao, S. Gao, and L. Chen, "Ship recognition method combined with image segmentation and deep learning feature extraction in video surveillance," *Multimedia Tools and Applications*, vol. 79, no. 13, pp. 9177–9192, 2020.

[74] Z. Dong and B. Lin, "Learning a robust CNN-based rotation insensitive model for ship detection in VHR remote sensing images," *International Journal of Remote Sensing*, vol. 41, no. 9, pp. 3614–3626, 2020.

[75] X. Chen, L. Qi, and Y. Yang, "Port ship detection in complex environments," in *Proceedings of the 2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*, pp. 1–6, IEEE, Piscataway, NJ, USA, August 2019.

[76] Z. Shao, L. Wang, and Z. Wang, "Saliency-aware convolution neural network for ship detection in surveillance video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 781–794, 2019.

[77] T. Liu, B. Pang, and S. Ai, "Study on visual detection algorithm of sea surface targets based on improved YoloV3," *Sensors*, vol. 20, no. 24, p. 7263, 2020.

[78] Z. Shao, W. Wu, and Z. Wang, "Seaships: a large-scale precisely annotated dataset for ship detection," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[79] H. Li, L. Deng, and C. Yang, "Enhanced Yolo v3 tiny network for real-time ship detection from visual image," *IEEE Access*, vol. 9, pp. 16692–16706, 2021.

[80] H. Huang, D. Sun, and R. Wang, "Ship target detection based on improved Yolo network," *Mathematical Problems in Engineering*, vol. 2020, Article ID 6402149, 10 pages, 2020.

[81] T. Cane and J. Ferryman, "Evaluating deep semantic segmentation networks for object detection in maritime surveillance," in *Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, Auckland, New Zealand, November 2018.

[82] S. Ghosh, P. K. Konugurthi, and G. Shankar Rao Singupurapu, "On-board ship detection for medium resolution optical sensors," *Sensors*, vol. 21, no. 9, p. 3062, 2021.

[83] L. Tian, Y. Cao, and B. He, "Image enhancement driven by object characteristics and dense feature reuse network for ship target detection in remote sensing imagery," *Remote Sensing*, vol. 13, no. 7, p. 1327, 2021.

[84] L. Chen, W. Shi, and D. Deng, "Improved YoloV3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images," *Remote Sensing*, vol. 13, no. 4, p. 660, 2021.

[85] Q. Wang, F. Shen, and L. Cheng, "Ship detection based on fused features and rebuilt YoloV3 networks in optical remote-sensing images," *International Journal of Remote Sensing*, vol. 42, no. 2, pp. 520–536, 2021.

[86] C. Cao, J. Wu, and X. Zeng, "Research on airplane and ship detection of aerial remote sensing images based on convolutional neural network," *Sensors*, vol. 20, no. 17, p. 4696, 2020.

[87] G. Tang, Y. Zhuge, and C. Claramunt, "N-Yolo: A SAR ship detection using noise-classifying and complete-target extraction," *Remote Sensing*, vol. 13, no. 5, p. 871, 2021.

[88] G. Tang, S. Liu, I. Fujino, and H. Yolo, "A single-shot ship detection approach based on region of interest preselected network," *Remote Sensing*, vol. 12, no. 24, p. 4192, 2020.

[89] R. Ribeiro, G. Cruz, and J. Matos, "A dataset for airborne maritime surveillance environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 9, pp. 2720–2732, 2017.

[90] G. Cruz and A. Bernardino, "Aerial detection in maritime scenarios using convolutional neural networks," in *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 373–384, Springer, Lecce, Italy, October 2016.

[91] C. D. Rodin, L. N. de Lima, and F. A. de Alcantara Andrade, "Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, July 2018.

[92] M. M. Marques, V. Lobo, and A. P. Aguiar, "An unmanned aircraft system for maritime operations: the automatic detection subsystem," *Marine Technology Society Journal*, vol. 55, no. 1, pp. 38–49, 2021.

[93] S. Xiu, Y. Wen, and H. Yuan, "A multi-feature and multi-level matching algorithm using aerial image and ais for vessel identification," *Sensors*, vol. 19, no. 6, p. 1317, 2019.

[94] G. Cruz and A. Bernardino, "Learning temporal features for detection on maritime airborne video sequences using convolutional LSTM," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6565–6576, 2019.

[95] S. Chen, R. Zhan, and W. Wang, "Learning slimming SAR ship object detector through network pruning and knowledge distillation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1267–1282, 2020.

[96] Z. Long, W. Suyuan, and C. U. I. Zhongma, "Lira-Yolo: a lightweight model for ship detection in radar images," *Journal of Systems Engineering and Electronics*, vol. 31, no. 5, pp. 950–956, 2020.

[97] R. Yang, Z. Pan, and X. Jia, "A novel CNN-based detector for ship detection based on rotatable bounding box in SAR images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1938–1958, 2021.

[98] F. S. Hass and J. Jokar Arsanjani, "Deep learning for detecting and classifying ocean objects: application of YoloV3 for iceberg–ship discrimination," *ISPRS International Journal of Geo-Information*, vol. 9, no. 12, p. 758, 2020.

[99] P. Chen, Y. Li, and H. Zhou, "Detection of small ship objects using anchor boxes cluster and feature pyramid network model for SAR imagery," *Journal of Marine Science and Engineering*, vol. 8, no. 2, p. 112, 2020.

[100] Y. Lang and B. Yuan, "Algorithm application based on the infrared image in unmanned ship target image recognition," *Microprocessors and Microsystems*, vol. 80, Article ID 103554, 2021.

[101] J. Xie, E. Stensrud, and T. Skramstad, "Detection-based object tracking applied to remote ship inspection," *Sensors*, vol. 21, no. 3, p. 761, 2021.

[102] F. Han, J. Yao, and H. Zhu, "Underwater image processing and object detection based on deep CNN method," *Journal of Sensors*, 2020.

[103] D. Lee, G. Kim, and D. Kim, "Vision-based object detection and tracking for autonomous navigation of underwater robots," *Ocean Engineering*, vol. 48, pp. 59–68, 2012.

[104] F. Farahnakian and J. Heikkonen, "Deep learning based multi-modal fusion architectures for maritime vessel detection," *Remote Sensing*, vol. 12, no. 16, p. 2509, 2020.
[105] M. Pan, Y. Liu, and J. Cao, "Visual recognition based on deep learning for navigation mark classification," *IEEE Access*, vol. 8, pp. 32767–32775, 2020.
[106] Y. Zhou, Y. Lu, and Y. Shen, "Polarized remote inversion of the refractive index of marine spilled oil from PARASOL images under sunglint," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 4, pp. 2710–2719, 2019.
[107] A. Martins, A. Dias, and J. Almeida, "Field experiments for marine casualty detection with autonomous surface vehicles," in *Proceedings of the 2013 OCEANS-San Diego*, pp. 1–5, IEEE, San Diego, CA, USA, September 2013.
[108] B. Wang, Y. Su, and L. Wan, "A sea-sky line detection method for unmanned surface vehicles based on gradient saliency," *Sensors*, vol. 16, no. 4, p. 543, 2016.
[109] C. Jeong, H. S. Yang, and K. D. Moon, "A novel approach for detecting the horizon using a convolutional neural network and multi-scale edge detection," *Multidimensional Systems and Signal Processing*, vol. 30, no. 3, pp. 1187–1204, 2019.
[110] D. K. Prasad, D. Rajan, and L. Rachmawati, "MuSCoWERT: consistence of weighted edge Radon transform for horizon detection in maritime images," *JOSA A*, vol. 33, no. 12, pp. 2491–2500, 2016.
[111] S. M. Ettinger, M. C. Nechyba, and P. G. Ifju, "Vision-guided flight stability and control for micro air vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617–640, 2003.
[112] S. Fefilatyev, D. Goldgof, and M. Shreve, "Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system," *Ocean Engineering*, vol. 54, pp. 1–12, 2012.
[113] H. Bouma, D. J. J. de Lange, and S. P. van den Broek, "Automatic detection of small surface targets with electro-optical sensors in a harbor environment," *International Society for Optics and Photonics*, vol. 7114, Article ID 711402, 2008.
[114] E. Gershikov, T. Libe, and S. Kosolapov, "Horizon line detection in marine images: which method to choose?" *International Journal on Advances in Intelligent Systems*, vol. 6, no. 1, 2013.
[115] B. A. Alpatov, P. V. Babayan, and N. Y. Shubin, "Weighted Radon transform for line detection in noisy images," *Journal of Electronic Imaging*, vol. 24, no. 2, Article ID 023023, 2015.
[116] C. Y. Jeong, H. S. Yang, and K. D. Moon, "Fast horizon detection in maritime images using region-of-interest," *International Journal of Distributed Sensor Networks*, vol. 14, no. 7, 2018.
[117] D. Liang and Y. Liang, "Horizon detection from electro-optical sensors under maritime environment," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 1, pp. 45–53, 2019.
[118] H. Zhang, P. Yin, and X. Zhang, "A robust adaptive horizon recognizing algorithm based on projection," *Transactions of the Institute of Measurement and Control*, vol. 33, no. 6, pp. 734–751, 2011.
[119] I. Lipschutz, E. Gershikov, and B. Milgrom, "New methods for horizon line detection in infrared and visible sea images," *International Journal of Computer Engineering Research*, vol. 3, no. 3, pp. 1197–1215, 2013.
[120] Y. Sun and L. Fu, "Coarse-fine-stitched: a robust maritime horizon line detection method for unmanned surface vehicle applications," *Sensors*, vol. 18, no. 9, p. 2825, 2018.
[121] W. Yang, H. Li, and J. Liu, "A sea-sky-line detection method based on Gaussian mixture models and image texture features," *International Journal of Advanced Robotic Systems*, vol. 16, no. 6, 2019.
[122] Y. Zhang, Q. Z. Li, and F. N. Zang, "Ship detection for visual maritime surveillance from non-stationary platforms," *Ocean Engineering*, vol. 141, pp. 53–63, 2017.
[123] X. Chen, L. Qi, and Y. Yang, "Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis," *Journal of Advanced Transportation*, vol. 2020, Article ID 7194342, 12 pages, 2020.
[124] Z. Li, L. Zhao, and X. Han, "Lightweight ship detection methods based on YoloV3 and DenseNet," *Mathematical Problems in Engineering*, vol. 2020, Article ID 4813183, 10 pages, 2020.
[125] D. Qiao, G. Liu, and J. Zhang, "M3C: multimodel-and-multicue-based tracking by detection of surrounding vessels in maritime environment for USV," *Electronics*, vol. 8, no. 7, p. 723, 2019.
[126] Z. Huang, B. Sui, and J. Wen, "An intelligent ship image/video detection and classification method with improved regressive deep convolutional neural network," *Journal of Complexity*, vol. 2020, Article ID 1520872, 11 pages, 2020.
[127] D. K. Prasad, C. K. Prasath, and D. Rajan, "Object detection in a maritime environment: performance evaluation of background subtraction methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1787–1802, 2018.
[128] Y. T. Chan, "Comprehensive comparative evaluation of background subtraction algorithms in open sea environments," *Computer Vision and Image Understanding*, vol. 202, Article ID 103101, 2021.
[129] J. M. Lee, K. H. Lee, and B. Nam, "Study on image-based ship detection for AR navigation," in *Proceedings of the 2016 6th International Conference on IT Convergence and Security (ICITCS)*, pp. 1–4, IEEE, Prague, Czech, September 2016.
[130] S. J. Leela, M. I. Roh, and M. J. Ohb, "Image-based ship detection using deep learning," *Ocean Systems Engineering*, vol. 10, 2020.
[131] S. T. Westlake, T. N. Volonakis, and J. Jackman, "Deep learning for automatic target recognition with real and synthetic infrared maritime imagery," *International Society for Optics and Photonics*, vol. 11543, Article ID 1154309, 2020.
[132] F. E. T. Schöller, M. K. Plenge-Feidenhans, and J. D. Stets, "Assessing deep-learning methods for object detection at sea from LWIR images," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 64–71, 2019.
[133] M. Nalamati, N. Sharma, and M. Saqib, "Automated monitoring in maritime video surveillance system," in *Proceedings of the 2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–6, IEEE, Wellington, New Zealand, November 2020.
[134] H. Feng, J. Guo, and H. Xu, "SharpGAN: dynamic scene deblurring method for smart ship based on receptive field block and generative adversarial networks," *Sensors*, vol. 21, no. 11, p. 3641, 2021.
[135] X. Hou, W. Ao, and Q. Song, "FUSAR-Ship: building a high-resolution SAR-AIS matchup dataset of Gaofen-3 for ship detection and recognition," *Science China Information Sciences*, vol. 63, no. 4, pp. 1–19, 2020.
[136] Y. You, Z. Li, and B. Ran, "Broad area target search system for ship detection via deep convolutional neural network," *Remote Sensing*, vol. 11, no. 17, 2019.

[137] M. Zhang, Y. Chen, and X. Liu, "Adaptive anchor networks for multi-scale object detection in remote sensing images," *IEEE Access*, vol. 8, pp. 57552–57565, 2020.

[138] Y. Zhuang, L. Li, and H. Chen, "Small sample set inshore ship detection from VHR optical remote sensing images based on structured sparse representation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2145–2160, 2020.

[139] L. Zhou, Z. Wang, and Y. Luo, "Separability and compactness network for image recognition and super-resolution," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3275–3286, 2019.

[140] Z. Xu, R. Hu, and J. Chen, "Semisupervised discriminant multimanifold analysis for action recognition," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 10, pp. 2951–2962, 2019.

[141] S. U. N. Xian, W. Zhirui, and S. U. N. Yuanrui, "AIR-SARShip-1.0: high-resolution SAR ship detection dataset," *Radar Journal*, vol. 8, no. 6, pp. 852–862, 2019.

[142] Y. Mao, Y. Yang, and Z. Ma, "Efficient low-cost ship detection for SAR imagery based on simplified U-net," *IEEE Access*, vol. 8, pp. 69742–69753, 2020.

[143] F. Gao, Y. He, and J. Wang, "Anchor-free convolutional network with dense attention feature aggregation for ship detection in SAR images," *Remote Sensing*, vol. 12, no. 16, p. 2619, 2020.

[144] S. Moosbauer, D. Konig, and J. Jakel, "A benchmark for deep learning based object detection in maritime environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA, USA, June 2019.

[145] C. Y. Jeong, H. S. Yang, and K. D. Moon, "Horizon detection in maritime images using scene parsing network," *Electronics Letters*, vol. 54, no. 12, pp. 760–762, 2018.

[146] R. Spraul, L. Sommer, and A. Schumann, "A comprehensive analysis of modern object detection methods for maritime vessel detection," *International Society for Optics and Photonics*, vol. 11543, Article ID 1154305, 2020.

[147] U. Ganbold and T. Akashi, "The real-time reliable detection of the horizon line on high-resolution maritime images for unmanned surface-vehicle," in *Proceedings of the 2020 International Conference on Cyberworlds (CW)*, pp. 204–210, IEEE, Caen, France, September 2020.

[148] V. Marié, I. Bechar, and F. Bouchara, *Towards Maritime Videosurveillance using 4K Videos*, Springer, Cham, Switzerland, pp. 123–133, 2018.

[149] C. Lin, W. Chen, and H. Zhou, "Multi-visual feature saliency detection for sea-surface targets through improved sea-sky-line detection," *Journal of Marine Science and Engineering*, vol. 8, no. 10, p. 799, 2020.

[150] V. Soloviev, F. Farahnakian, and L. Zelioli, "Comparing cnn-based object detectors on two novel maritime datasets," in *Proceedings of the 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, London, UK, July 2020.

[151] V. Marie, I. Bechar, and F. Bouchara, "Real-time maritime situation awareness based on deep learning with dynamic anchors," in *Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, Auckland, New Zealand, November 2018.

[152] S. Li, Z. Zhang, and B. Li, "Multiscale rotated bounding box-based deep learning method for detecting ship targets in remote sensing images," . *Sensors*, vol. 18, no. 8, p. 2702, 2018.

[153] P. Spagnolo, F. Filieri, and C. Distante, "A new annotated dataset for boat detection and re-identification," in *Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–7, IEEE, Taipei, Taiwa, September 2019.

[154] B. Bovcon, J. Muhovič, and J. Perš, "The mastr1325 dataset for training deep usv obstacle detection models," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3431–3438, IEEE, Venetian Macao, Macau, November 2019.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5538296, 12 pages
https://doi.org/10.1155/2021/5538296

Research Article

Speed Proportional Integrative Derivative Controller: Optimization Functions in Metaheuristic Algorithms

Luis Fernando de Mingo López,[1] **Francisco Serradilla García**,[1,2] **José Eugenio Naranjo Hernández**,[1,2] **and Nuria Gómez Blas**[1]

[1]*ETSI de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, Spain*
[2]*Instituto Universitario de Investigación del Automóvil (INSIA), Universidad Politécnica de Madrid, Madrid, Spain*

Correspondence should be addressed to José Eugenio Naranjo Hernández; joseeugenio.naranjo@upm.es

Received 20 January 2021; Revised 7 July 2021; Accepted 2 October 2021; Published 3 November 2021

Academic Editor: Chi-Hua Chen

Recent advancements in computer science include some optimization models that have been developed and used in real applications. Some metaheuristic search/optimization algorithms have been tested to obtain optimal solutions to speed controller applications in self-driving cars. Some metaheuristic algorithms are based on social behaviour, resulting in several search models, functions, and parameters, and thus algorithm-specific strengths and weaknesses. The present paper proposes a fitness function on the basis of the mathematical description of proportional integrative derivate controllers showing that mean square error is not always the best measure when looking for a solution to the problem. The fitness developed in this paper contains features and equations from the mathematical background of proportional integrative derivative controllers to calculate the best performance of the system. Such results are applied to quantitatively evaluate the performance of twenty-one optimization algorithms. Furthermore, improved versions of the fitness function are considered, in order to investigate which aspects are enhanced by applying the optimization algorithms. Results show that the right fitness function is a key point to get a good performance, regardless of the chosen algorithm. The aim of this paper is to present a novel objective function to carry out optimizations of the gains of a PID controller, using several computational intelligence techniques to perform the optimizations. The result of these optimizations will demonstrate the improved efficiency of the selected control schema.

1. Introduction

Many optimization problems are nondeterministic polynomial-time (NP) or NP-hard, and a high computing power is required when trying to solve them [1, 2]. An NP-hard problem "is a problem where a solution for it is at least as hard as finding a solution for the hardest problem whose solution can quickly be checked as being true. Some NP-hard problems are ones in which a working solution can be checked quickly (NP problems) and some are not. NP-hard problems are also NP problems fit into a label called NP-complete" [3]. Many of the problems arising in present-day applications from scientific fields belong in NP-hard problems: they involve search spaces with many dimensions, they are multimodal or multiobjective, and the optimization functions are hard to compute or are applied on large volumes of data. Classical optimization methods from operation research make it possible to find optimal solutions for complex problems, but are not useful in practice due to their excessive computational load when applied to real-world systems [4]. For NP-hard problems, the time required to solve a problem grows exponentially with respect to the size of the problem, making the exact methods unpractical.

In order to solve the disadvantages of classical trial-and-error methods and mathematical solution search techniques, researchers have proposed various algorithms that mimic natural and artificial phenomena, and black-box optimization benchmark problems are implemented to evaluate the performance of these algorithms.

The research area of metaheuristic search/optimization algorithms has an active development of finding inspiration in nature, especially in social behaviour. Among the most classical models are genetic algorithms [5–7], but there are others based on the social behaviour of ants and ant colony optimization [8–10]; in recent years, a lot of methods based on natural heuristics have been proposed: birds [11, 12], reincarnation [13], zombies [14], bees [15–17], etc. This paper shows the application of 21 algorithms' implementation as follows: particle swarm optimization (PSO) [18], clonal selection algorithm (CLONALG) [19], whale optimization algorithm (WOA) [20], shuffled frog leaping (SFL) [21], differential evolution (DE) [22, 23], cat swarm optimization (CSO) [24], ant lion optimizer (ALO) [25], artificial bee colony algorithm (ABC) [26], firefly algorithm (FFA) [27], grey wolf optimizer (GWO) [28], dragonfly algorithm (DA) [29], grasshopper optimization algorithm (Goa) [30], genetic algorithm (GA) [31], harmony search algorithm (HS) [32], sine cosine algorithm (SCA) [33], moth flame optimizer (MFO) [34], krill herd algorithm (KH) [35], bat algorithm (BA) [36], gravitational-based search (GBS) [37], cuckoo search (CS) [38], and black hole optimization (BHO) [39].

Some of the most representative computational intelligence algorithms today are earthworm optimization algorithm (EWA) [40], monarch butterfly optimization (MBO) [41, 42], moth search (MS) algorithm [43], slime mould algorithm (SMA) [44], elephant herding optimization (EHO) [45], and Harris hawks optimization (HHO) [46], among others [47, 48].

Each of these algorithms has its own peculiarities, and they can be used to solve various categories of optimization problems. The performance of evolutionary algorithms depends on the choice of their control parameters, which must be tuned for each specific problem. Population-based metaheuristic algorithms generally include two processes: exploration and exploitation. The exploration process attempts to analyse a wide variety of solutions and must be random enough to ensure that it covers a large area of the problem space. The exploitation process tries to improve, through a local search, the solutions found in the exploration process. In this phase, the optimizer focuses on the neighborhood of the highest quality solutions found by the exploration process, rather than the entire solution space. In any optimization process, it is essential to find an adequate balance between exploration and exploitation, a balance that depends on the nature of the specific problem to be solved.

Hybrid optimization techniques are available in the state of the art for solving complex engineering problems, which combine several simple algorithms to obtain better optimization efficiency.

In the automotive industry, especially for vehicle design and optimized components, the Butterfly Optimization Algorithm (BOA) is a widely used hybrid optimization technique. Also, other hybridization proposals are available for similar purposes, such as Harris Hawk optimization with simulated annealing (HHOSA), which provides an accelerated convergence. Additionally, a hybridization of the interior search algorithm with the hill climbing algorithm (H-ISA) has been used to optimize structural and mechanical design problems, among others. This algorithm has been proven to work better in those cases than the ant lion optimizer (ALO), gravitational search algorithm (GSA), firefly algorithm (FFA), league championship algorithm (LCA), bat algorithm (BAT), symbiotic imperialist competitive algorithm (ICA), organisms search (SOS), and charged system search algorithm (CSS).

The key element for any of these algorithms is the fitness function, which must meet the following conditions: (a) the function must be clearly defined; (b) the implementation of fitness must be as efficient as possible; since it will be evaluated many times in the optimization process, its efficiency has a strong impact on the overall performance of the algorithm; (c) fitness must provide a quantitative measure of the quality of a given solution; (d) the best individuals must produce the best values of fitness function and vice versa.

The fitness function is dependent on the problem domain. The implementation of this fitness function is the main part when you want to apply a nature-based algorithm to a problem, and it must be built for each specific problem. For certain types of problems, for example, for classification tasks with supervised learning, error metrics, such as the Euclidean distance or the Manhattan distance, are often used. For other optimization problems, the summation of a set of quality indicators related to the application domain can be used.

This paper shows the importance of choosing a right fitness function; it is a key point to get a good performance regardless of the chosen algorithm. Discussion about whether a given algorithm obtains a better result than others is not addressed in this study.

The main highlights of the paper are

(i) The evaluation of different metaheuristics techniques applied to PID controllers

(ii) Showing the importance of choosing the right fitness function (mean squared error is not always the best choice)

(iii) Final fitness function that improves the mean squared error achieving a quasi-perfect fit of desired signal

2. Proportional Integrative Derivative Controller

Proportional integrative derivative controllers are the controllers par excellence, as well as the most studied. This type of controllers is based on the old proportional controls, such as the centrifugal regulator of Watt's machine (1788). One important thing is that knowledge about the process to control is not really needed when using it with a PID controller, which is the reason why such a mathematical model is so broadly used. The PID equation describes how to control the system: it receives an error calculated from the desired/obtained output, using this error to feed the control loop in the next iteration. The aim of the controller is to minimize this calculated error by adjusting the gains of the PID.

The PID controller is based on the relation of three components: the proportional, the integral, and the derivative. The tune of the gains of these components not only generates the behaviour of the output of the controller but also influences the others. Although those parameters are tuned, it is quite difficult to manually define the values of the PID gains that generate the values tending to zero of the performance metrics (establishment times, over-oscillation, settling, etc.). However, the best tuning of the controller tries to get a trade-off among those error metrics, adjusting them according to the requirements of the application.

A PID controller outputs a signal using a weighted sum of three terms:

(i) Proportional: error between the defined setpoint and the actual process value

(ii) Integral: how much error has been accumulated over time

(iii) Derivative: how rapidly the error is changing

This naturally leads to P (Proportional), I (Integral), and D (Derivative) in PID and the three weighting parameters (K_p, K_i, K_d) that need to be computed. By modifying these control parameters, the behaviour of these on the system will be modified, and it will be possible to obtain a controller that satisfies the design criteria established for a specific application.

Each control action is responsible for producing a specific effect on the transitional or permanent regime, or on both at the same time. Therefore, when it comes to achieving better control over the system, it is necessary to know which aspect of the control is to be improved, to modify a specific action and not to disrupt the rest of the system.

One must be aware that proportional action affects both the transitional and the permanent regime, tending to reduce the error in the permanent regime when its action increases. On the contrary, when this action increases the system will tend to increase the oscillations in the process variable.

The integral action affects the permanent regime, annulling the error. The error could be understood as the difference between the reference and the value read by the sensor. If the error is greater than zero, this means that the value read by the sensor will be below the reference, meaning that the control action will increase, while if the error is less than zero, the value read by the sensor will be greater than the reference, and therefore, the action on the system will decrease.

The last control action is the derivative action. This action is mainly focused on the transitional regime, improving the stability of the system. It can also be found in some articles as a predictive action, and this is because this control action is of an anticipatory nature, i.e., it anticipates the behaviour of the system to improve its performance. The main disadvantage of this action is that it amplifies the noise of the signal and saturates the actuators in the event of sudden changes in the set point.

When designing a PID controller, several methods can be found in the literature. One of the most famous methods is the Ziegler–Nichols method [49], which is proposed by Ogata [50]. Other methods can also be found, such as those Chien–Hrones–Reswick methods proposed by [51] and the Cohen-Coon method proposed by [52]. In order to tune the PID controller, using these methods, it will be necessary to know the behavior of the system. To do this, the corresponding actuator will be put into operation and wait until the variable in question is stabilized in a permanent state. Observing the variable to control shape (see Figure 1), the values of the gain K, the delay L, and the time constant T (with $a = L/T$) could be calculated depending on the chosen method (see Tables 1–4).

Some authors focus on improving the response of the PID controllers using metaheuristics, especially in problems with a highly nonlinear behaviour, because, for them, the traditional methods of adjusting the gains are inefficient, obtaining results comparable or superior to conventional techniques [53]. Valluru and Singh [54] show that the efficiency of tuning nonlinear drivers using particle swarm optimization and other bioinspired techniques is superior to the results with traditional tuning techniques. The experimental results show that the overshoot and settling time of a nonlinear PID controller can be improved while maintaining a satisfactory system response. Recent research [55] shows that the performance of a PID control system with gains calculated by a symbiotic organism search algorithm manages to minimize the steady-state error. When the system is subjected to disturbances, this controller is capable of reaching the set point in any situation.

This paper does not take disturbance into account; it only shows the importance of choosing a right fitness function. Talking about exogenous disturbance (input disturbance) that affects model (physical or mathematical) behaviour is most probably assimilated to account for model-uncertainty (inaccuracy), time-varying parameters, perturbation (as wind-gust in aircrafts), actuator unmodelled dynamics, and the same ilk. Input disturbance could be modelled as a constant signal $D(s) = 1/s$, in Laplace domain, or $d(t) = 1$ in time domain, or a sinusoidal signal, or a stochastic process, or something like that.

3. Problem Description

There is an electric vehicle with cameras and sensors for speed, acceleration, LiDaR, etc. (see Figure 2). This vehicle can be driven remotely through the ODBII port (see Figure 3) by adjusting the cruising speed, acceleration, and turning the steering wheel. The problem is to set a cruising speed (higher or lower than the current one) and see the vehicle's response under real conditions.

The cruise control is based on a PID. A manual adjustment of the parameters (K_p, K_i, K_d) could be done, but the idea is to try to find a better solution than the manual operator or expert could achieve. The problem can be formulated as a model of optimization in a three-dimensional space. The optimization process requires an objective function (fitness) to be minimized depending on the parameters (K_p, K_i, K_d). Mean square error is the classical

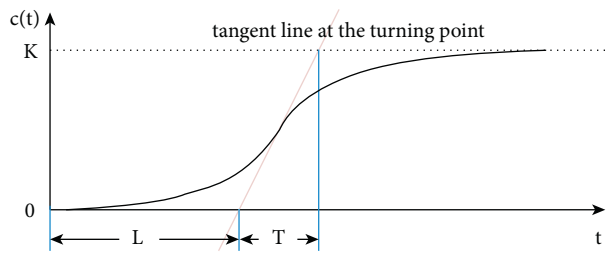


Figure 1: How to compute values corresponding to the gain K, the delay L, and the time constant T over the signal to control. Different methods can be applied to obtain (K_p, K_i, K_d) parameters of the controller (see Tables 1–4).

Table 1: The Ziegler–Nichols tuning rules create a quarter wave decay. This is an acceptable result for some purposes, but not optimal for all applications [49, 50].

Controller	k_p	k_i	k_d
P	$1/a$	0	0
PI	$0.9/a$	$3L$	0
PID	$1.2/a$	$2L$	$L/2$

Table 2: The Chien–Hrones–Reswick autotuning method focuses on set point response and disturbance response (20% overshoot) [51].

Controller	k_p	k_i	k_d
P	$0.3/a$	0	0
PI	$0.6/a$	$4L$	0
PID	$0.95/a$	$2.4L$	$0.42L$

Table 3: The Chien–Hrones–Reswick tuning rules (20% overshoot) [51].

Controller	k_p	k_i	k_d
P	$0.7/a$	0	0
PI	$0.7/a$	$2.3L$	0
PID	$1.2/a$	$2L$	$0.42L$

Table 4: The Cohen-Coon tuning rules (0% overshoot). Note that these rules produce a quarter-amplitude damping response [52].

Controller	k_p	k_i	k_d
P	$1/a(1 + (0.35\tau/1 - \tau))$	0	0
PI	$0.9/a(1 + (0.92\tau/1 - \tau))$	$L(3.3 - 3\tau/1 + 1.2\tau)$	0
PD	$1.24/a(1 + (0.13\tau/1 - \tau))$	0	$L(0.27 - 0.36\tau/1 - 0.87\tau)$
PID	$1.35/a(1 + (0.18\tau/1 - \tau))$	$L(2.5 - 2\tau/1 + 0.39\tau)$	$L(0.37 - 0.37\tau/1 - 0.18\tau)$

Figure 2: Autonomous electric vehicle used in real testing and light detection and ranging sensor (LiDAR) placed on the vehicle windscreen.

fitness function in optimization problems, but there are other approaches depending on the problem to solve. A good performance can be obtained, when dealing with PID, using a linear combination of overshoot, decay, setting time, and steady-state error as proposed by [56, 57]. This paper uses the same mathematical model of the system/process for which the controller described in [56, 57] is employed.

A fractional-order PID controller has better control performance than classical PID controllers. A fractional-order controller includes the capacity of adjusting responses

Figure 3: ODBII interface for controlling the vehicle and visualization of front and rear cameras for remotely controlling speed and direction.

of the control system in time and frequency. This feature grants a better and more robust performance than classical PID controllers. It is not always the case in which a fractional-order PID controller used for integer-order plants will be better than an integer-order PID. However, it has been shown in [58, 59] that the use of a fractional-order PID could make the entire control system perform better. This is because using a fractional one could satisfy 5 robustness criteria $(K_p, K_i, K_d, \mu, \delta)$ at most as compared to the usual classical PID controller which only has 3 parameters to be tuned for 3 robustness criteria. This paper uses a PID controller with precalculated μ and δ values from [56, 57].

The first fitness function f_1, see equation (1), is based on the mean square error (MSE), that is, the average squared difference between the estimated/target speed signal and the actual speed signal:

$$f_1 = \sqrt{\sum_t (\text{target}(t) - \text{signal}(t))^2}. \tag{1}$$

The second fitness function f_2, see equation (2), uses a lineal combination of the overshoot and decay of the signal to avoid large peaks and oscillation in the control response, with precomputed values of α and β [56, 57]:

$$f_2 = \alpha\Omega + \beta\Delta. \tag{2}$$

And the third one f_3, see equation (3), uses a lineal combination of the overshoot, decay, setting time, and steady-state error of the signal to avoid large peaks and oscillation in the control response and get an almost fit response to the original signal:

$$f_2 = \alpha\Omega + \beta\Delta + \gamma\Gamma + \theta\Theta. \tag{3}$$

The overshoot Ω can be defined as the gap between the outputs of the controller from the set point steady-state value. Following [50], a controller gain tuning is considered good in the case of an overdamped response with a minimum overshoot, according to the application. The value of the overshoot, see (4), is the difference between the system output and the target value (with $\alpha = 10$ and $\beta = 0.9$, to obtain better results):

$$\Omega = \begin{cases} \max(\text{signal}(t)) - \text{target}, & \text{when } e(t) > 0, \\ \text{target} - \min(\text{signal}(t)), & \text{when } e(t) < 0. \end{cases} \tag{4}$$

The decay ratio Δ is the value between two consecutive maxima of the controller output for a step change in the set point. This ratio is shown as

$$\Delta = \frac{c}{a}, \tag{5}$$

where the parameter a is the amplitude of the oscillations in the instant $t - 1$ and c the amplitude of the oscillation in the time t.

Let Γ be the settling time, that is, the time required for the controller output to stabilize at p% of the set point. According to literature, the recommended value of p is 0.02% [56, 57]:

$$\Gamma = \frac{t}{T_{\max}}, \quad \text{when } |\text{signal}(t) - \text{signal}(t-1)| < p, \tag{6}$$

where the parameter t is the amount of time in which the output signal is below p and $T_{\max}$ is the maximum value of the sampling time, considering the signal as the output value of the system.

The steady-state error Θ is the gap between the stabilized output signal of the system and the set point, after N controller iteration. In this paper, the selected value of N is 350, time enough for the controller to reach the set point (target):

$$\Theta = |\text{signal}(N) - \text{target}|. \tag{7}$$

Figure 4 shows the desired (blue) signal and the estimated/target (red) signal using the parameters computed by a particle swarm optimization algorithm: $(K_p = 14.1300, K_i = 4.6007, K_d = 0.00100000)$ with fitness f_1 value of 48.47595, $(K_p = 10.899968, K_i = 51.6346372, K_d = 0.001000000)$ with fitness f_2 value of 103.0716, and $(K_p = 10.901181, K_i = 100.000\,0000, K_d = 0.001000000)$ with fitness f_3 value of 26.21848. It can be seen that, in Figure 4(b), the peaks are smaller and the changes in signal are not as abrupt as in Figure 4(a), but Figure 4(c) shows the best results if using the full fitness function with all terms (overshoot, decay, setting time, and steady-state error), which is the expected and desired behaviour of the PID controller. Figure 4(c) is equivalent to the classical computation of PID.

4. Results

The particle swarm optimization algorithm has shown good results, see Figure 4, combined with an objective function f_3.

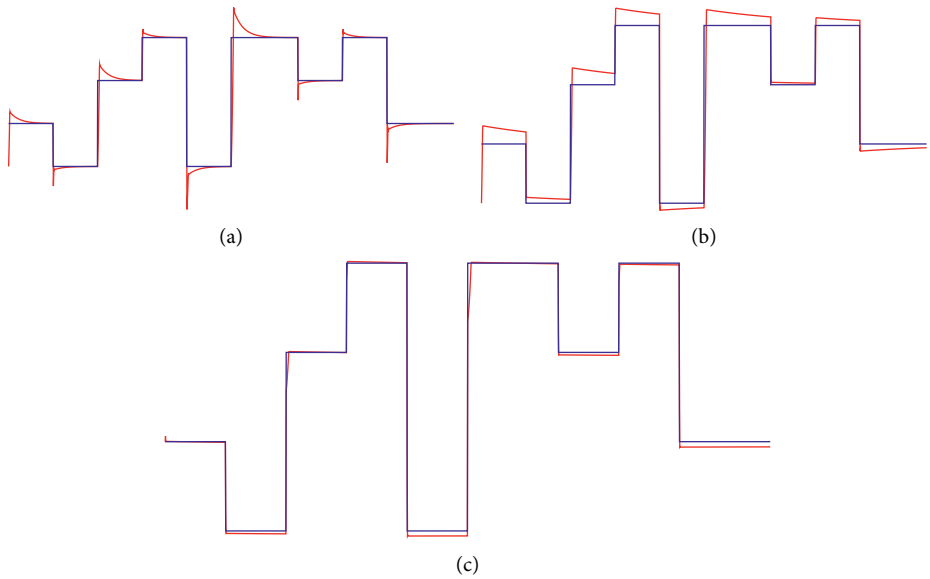


Figure 4: Optimization results using a PSO algorithm with 200 iterations and a population size of 20. Mean square error fitness f_1: (a) equation (1) vs. custom fitness f_2, (b) equation (2) vs. improved custom fitness f_3, and (c) equation (3). (a) Mean square error (equation (1)). (b) Overshoot and decay (equation (2)). (c) Overshoot, decay, setting time, and steady-state error as fitness function (equation (3)).

This section shows the results obtained using other metaheuristics (particle swarm optimization, whale optimization algorithm, differential evolution, clonal selection algorithm, cat swarm optimization, artificial bee colony algorithm, shuffled frog leaping, grey wolf optimizer, ant lion optimizer, firefly algorithm, dragonfly algorithm, harmony search algorithm, genetic algorithm, grasshopper optimization algorithm, sine cosine algorithm, moth flame optimizer, cuckoo search, bat algorithm, gravitational based search and black hole optimization, and krill herd algorithm) and the proposed three fitness functions f_1, f_2, and f_3. Optimization parameters have been set to the same values for all algorithms, 100 iterations, and a population size of 15 individuals with $(K_p, K_i, K_d) \in [0.001, 100]$, in order to get a similar testing environment for all algorithms.

The first approach consists in using the mean square error as a fitness function (see Table 5, equation (1), and Figure 5). In this case, some algorithms do not get a right solution since the output signal has a great oscillation pattern, but the ones that get good results produce peaks in the step function. This output is similar to the classical PID mathematical computation [49–51]. This behaviour could be considered a valid solution in other environments/problems but not in the case of a self-driving car with autonomous or remote speed controller as there are peaks and the car speed is not controlled in the transitions.

The second approach consists in using the overshoot and decay parameters (see Table 6, equation (2), and Figure 6). Now, the results are better than in the previous case since peaks disappear and the speed is controlled in a less abrupt mode (smooth transitions). As in the other case, some algorithms present oscillations, which is due to the lack of iterations or population size.

Finally, fitness function f_3, with the lineal combination of overshoot, decay, setting time, and steady-state error, is tested against the metaheuristics (see Table 7, equation (3), and Figure 7). The figure shows that the output signal fits the desired signal with no peaks and with an almost exact matching.

The main contribution of this paper is the use of the weighted fitness function definition f_3, as well as the use of this function as the basis to perform a metaheuristics optimization of the gains of a PID controller. The results of classical PID optimization usually generate controllers with an oscillating output. However, this oscillating output can be corrected using a low-pass filter, taking into account that the performance criteria always have less quality than metaheuristics optimization.

This paper has presented the results of metaheuristic optimization algorithms, based on different error metrics (1)–(3). These equations define the error as a lineal combination of the four estimators explained above. This error is calculated for a set point sequence instead of for a single set point; therefore, the controller obtained is more robust in new situations. In addition, the error is calculated over a different sequence than that used in the tuning, so the estimation of the controller error is more realistic.

Using these defined fitness functions and a randomly generated sequence of set points, a set of different gains has been obtained for the PID speed controller of an autonomous vehicle through a series of metaheuristics-based optimization techniques.

With most of the optimization algorithms studied, the ground truth error is improved. However, CS, CSO, and MFO perform worse than this ground truth. This fact could be due to the existence of multiple local minima in the parameter space for PID controller gains, which means that local search algorithms show premature convergence. In particular, the MFO algorithm finds different optima in some executions, seemingly indicating the existence of multiple local minima.

Table 5: Optimization solution values using proposed metaheuristics, using fitness f_1, see equation (1) and Figure 5.

Algorithm	K_p	K_i	K_d	Fitness
ABC	12.469341	4.48158890	0.00100000	48.50611
ACO	16.046610	35.68476668	0.00100000	51.17129
ALO	12.146504	4.21416994	0.00100000	48.49356
BA	5.093248	2.85260861	4.72422839	202.35233
BHO	1.734270	0.07296765	5.29040061	78.61255
CLONALG	32.463294	61.55373554	0.54027825	239.91586
CS	**14.432591**	**12.47369880**	**0.00100000**	**49.47551**
CSO	14.130092	4.60077827	0.00100000	48.47595
DA	14.130092	4.60078172	0.00100000	48.47595
DE	41.961478	1.82844085	20.91916782	490.12491
FFA	2.556885	1.04583741	0.55790335	71.08043
GA	17.332015	5.53366128	0.00100000	49.89310
GBS	27.031364	79.78306112	0.00100000	63.04356
GOA	14.132092	4.58780177	0.00100000	48.47596
HS	5.162984	1.35881040	1.38566380	84.41693
KH	17.009334	0.95704153	0.08686292	75.33240
MFO	14.130088	4.60078202	0.00100000	48.47595
PSO	26.413843	33.36398724	0.00100000	62.35269
SCA	14.209402	4.83498450	0.00100000	48.47908
SFL	5.972453	14.85325707	2.00940154	206.63336
WOA	14.128297	4.56045037	0.00100000	48.47605

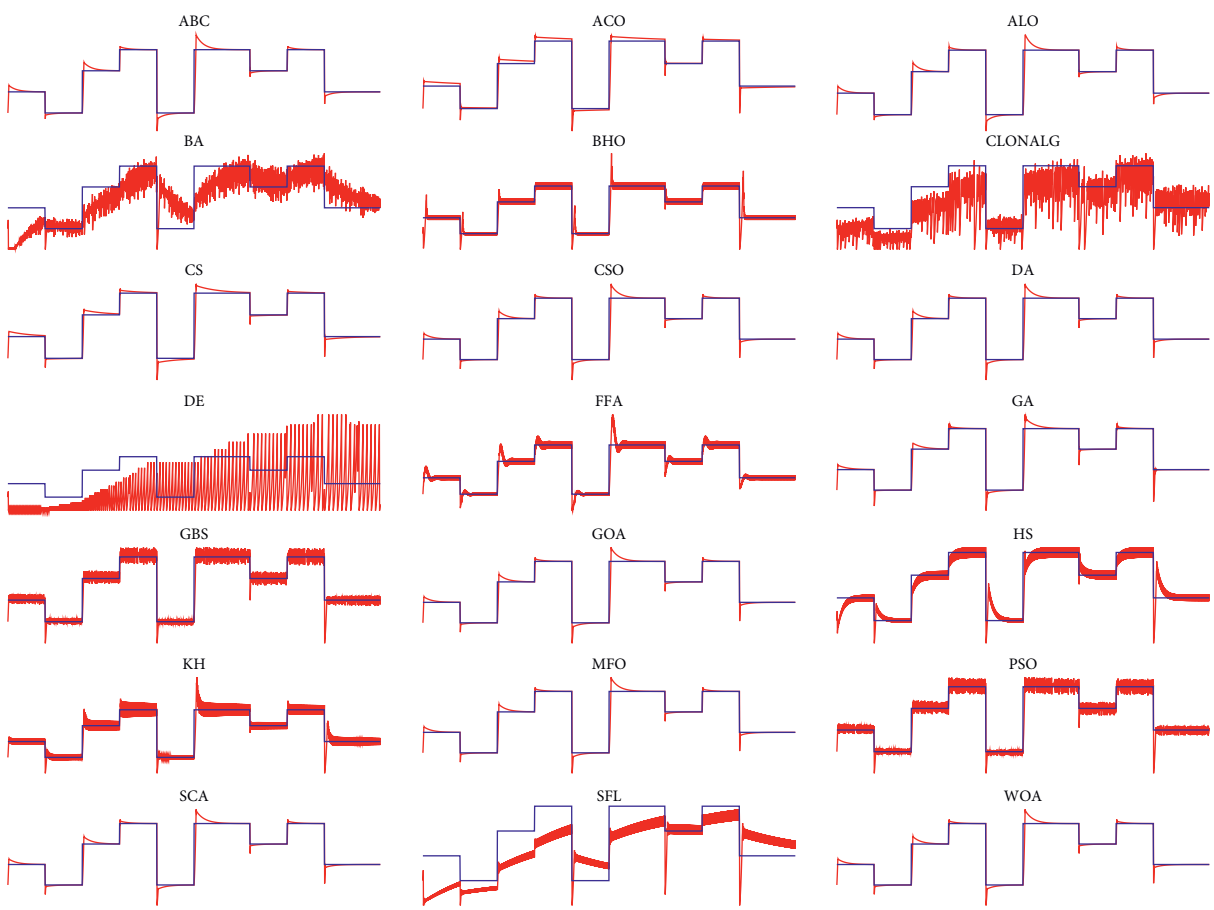


Figure 5: Results of 21 optimization metaheuristics using the mean square error as the fitness/objective function f_1: the red line is target/output signal and the blue line is desired signal, see equation (1) and Table 5. Parameters have been set to the same values for all algorithms, 100 iterations, and a population size of 15 individuals with $(K_p, K_i, K_d) \in [0.001, 100]$, in order to get a similar testing environment for all algorithms.

Table 6: Optimization solution values using proposed metaheuristics, using fitness f_2, see equation (2) and Figure 6.

Algorithm	K_p	K_i	K_d	Fitness
ABC	1.247355	29.8497584	2.024345564	127.5915
ACO	10.900608	64.1849888	0.001000000	103.1267
ALO	10.030650	13.1977015	0.001000000	108.4750
BA	60.898153	24.6557423	3.061338914	169.5129
BHO	6.748417	51.3241341	0.068173507	111.4488
CLONALG	4.201685	55.5608255	0.011870125	127.6016
CS	10.863056	13.4336943	0.001000000	107.0472
CSO	10.901259	100.0000000	0.001000000	103.5151
DA	10.900160	51.8360956	0.001000000	103.0721
DE	47.586536	34.9134975	3.302378887	183.9771
FFA	2.910704	35.4693957	1.001861920	129.7192
GA	10.847629	34.3291956	0.001000000	103.3316
GBS	0.417065	0.8933817	52.958178733	128.2626
GOA	10.899268	48.2122536	0.001000758	103.0773
HS	21.376061	1.3812947	49.333388968	181.5794
KH	60.249693	11.1017443	25.715143616	249.9684
MFO	**10.900022**	**51.4431324**	**0.001000000**	**103.0715**
PSO	10.899968	51.6346372	0.001000000	103.0716
SCA	10.876744	46.7826595	0.001032572	103.1077
SFL	75.568124	23.1263233	58.945069279	263.1667
WOA	10.823591	62.7439498	0.001796744	103.2075

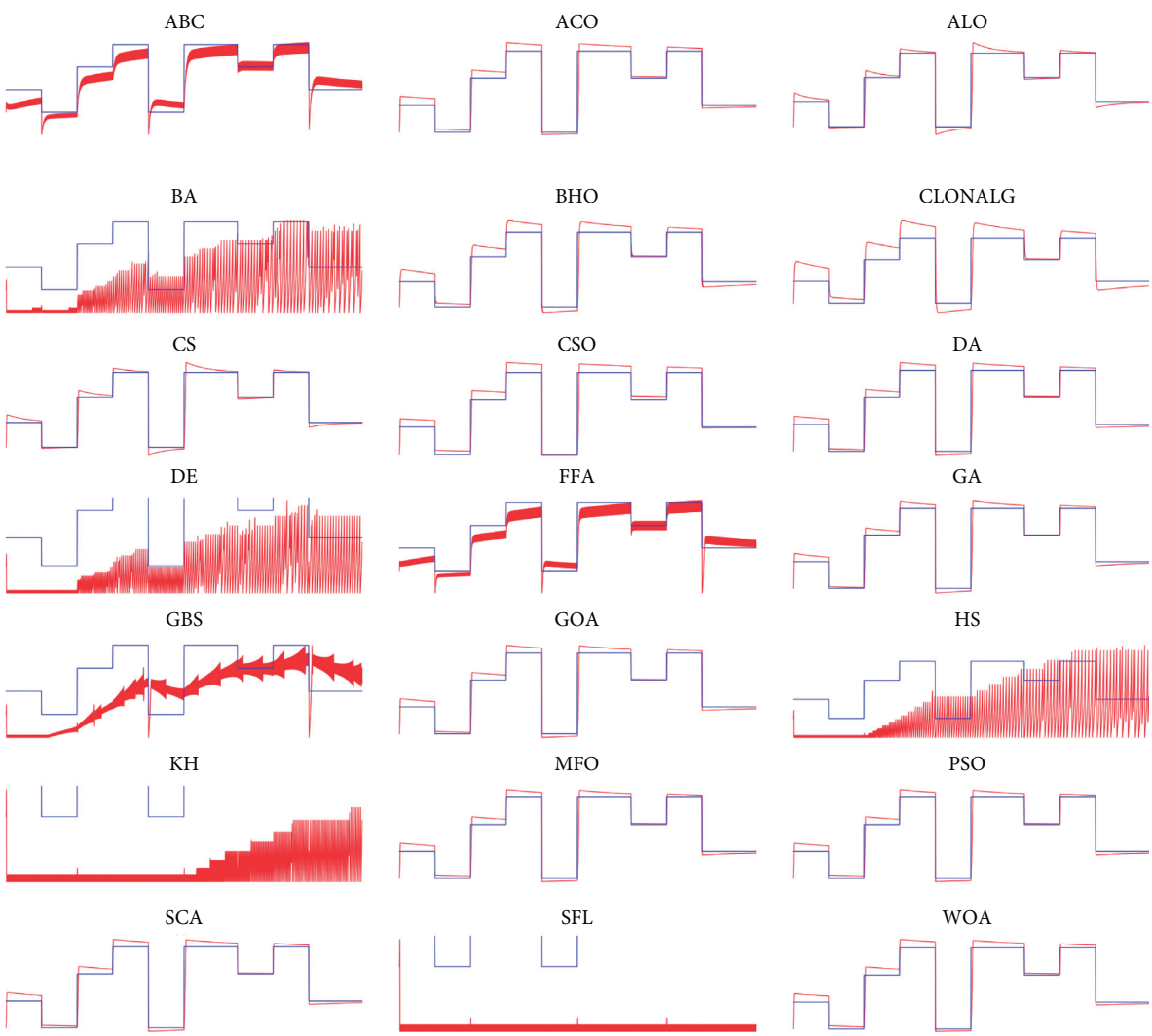


Figure 6: Results of 21 optimization metaheuristics using a fitness/objective function f_2 with overshoot and decay components: the red line is target/output signal and the blue line is desired signal, see equation (2) and Table 6. Parameters have been set to the same values for all algorithms, 100 iterations, and a population size of 15 individuals with $(K_p, K_i, K_d) \in [0.001, 100]$, in order to get a similar testing environment for all algorithms.

Table 7: Optimization solution values using proposed metaheuristics, using fitness f_3, see equation (3) and Figure 7.

Algorithm	K_p	K_i	K_d	Fitness
ABC	4.317014	100.0000000	0.414558038	38.13464
ACO	10.899466	44.7705683	0.001000000	27.30494
ALO	10.465441	29.5566926	0.001000000	28.41234
BA	1.380994	10.2406319	2.940162854	50.77700
BHO	1.215425	99.6389347	1.649872970	38.02265
CLONALG	16.254893	88.6727789	0.319180483	51.81580
CS	9.001000	9.0010000	0.001000000	33.31014
CSO	**10.901176**	**100.0000000**	**0.001000000**	**26.21848**
DA	2.181195	16.8407815	0.924729639	37.82754
DE	76.884508	65.8029157	0.692211458	74.54958
FFA	9.095726	99.9275577	0.019747615	27.25296
GA	100.000000	27.9241829	1.851140432	77.36712
GBS	32.971795	68.7382582	0.001000000	44.95544
GOA	10.801275	100.0000000	0.001930287	26.26413
HS	6.971770	3.4059226	1.211073046	45.29076
KH	8.067583	12.4642292	2.274522387	52.22566
MFO	**10.901181**	**100.0000000**	**0.001000000**	**26.21848**
PSO	27.551768	25.0129883	35.913471627	134.51197
SCA	10.775543	100.0000000	0.001000000	26.27785
SFL	2.687796	3.1262379	1.112656001	39.43179
WOA	61.659377	0.6926213	76.733240892	84.92107

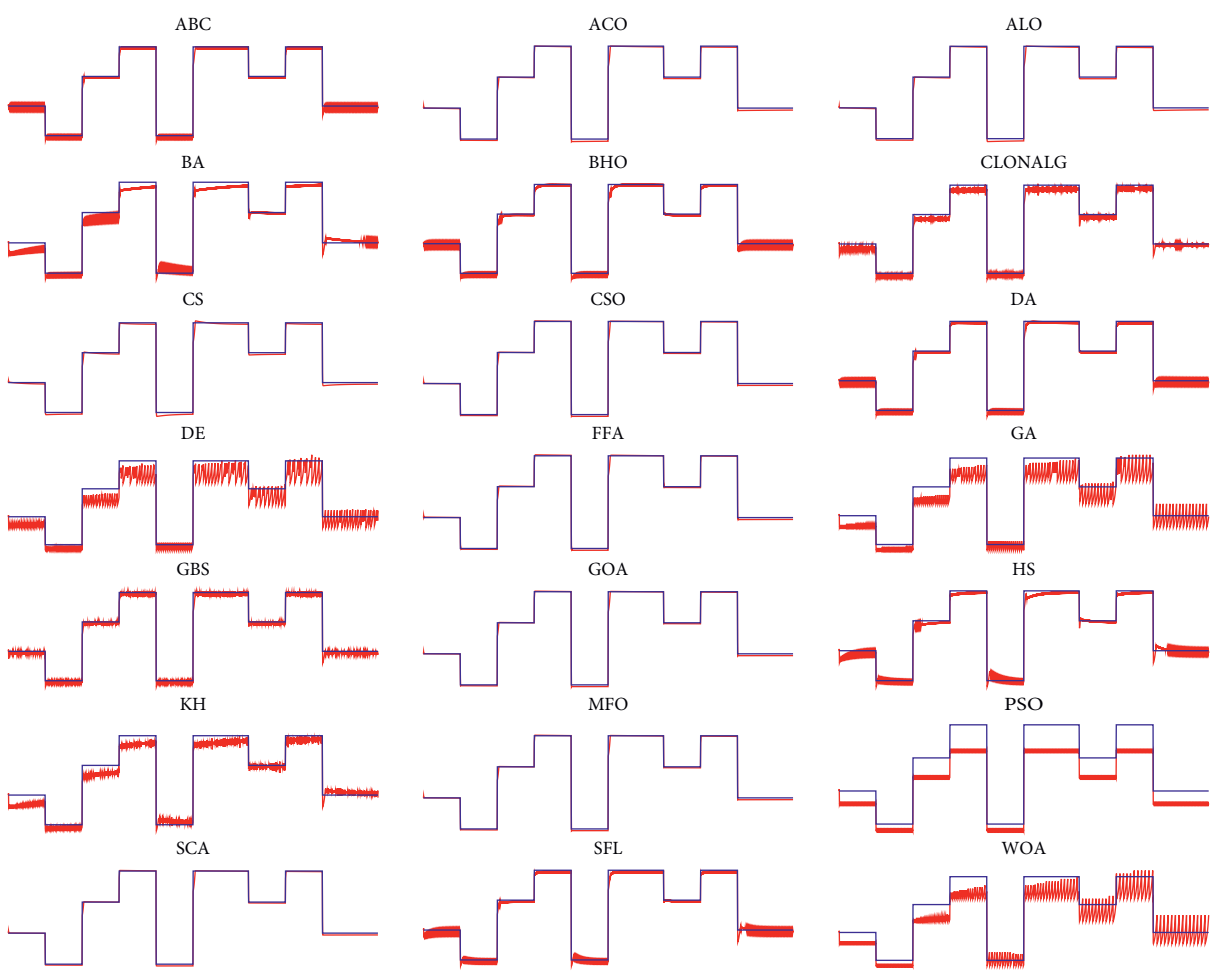


Figure 7: Results of 21 optimization models using a fitness/objective function f_3 with overshoot, decay, setting time, and steady-state error components: the red line is target/output signal and the blue line is desired signal, see equation (3) and Table 7. Parameters have been set to the same values for all algorithms, 100 iterations, and a population size of 15 individuals with $(K_p, K_i, K_d) \in [0.001, 100]$, in order to get a similar testing environment for all algorithms.

5. Conclusions

This paper shows the performance of the computation power of several evolutionary algorithms to optimize the gains (K_p, K_i, K_d) of the speed PID controller of a self-driving vehicle. Three fitness functions have been used to optimize the controller gains. Those functions are based in a combination of classic error metrics of the controller: overshoot, decay ratio, settling time, and steady-state error. The results of the different optimizations of the PID speed controller gains have been compared with each other, both for the sequence of step set points uses as the target of the optimization and for a totally new randomly generated sequence, unknown by the system. This system has been implemented by means of a dynamic model of the vehicle in simulation that represents the speed behaviour in a realistic way.

The fitness function proposed to measure the quality of a PID controller according to the traditional PID quality metrics reported by the literature, meaning that the minimization of this fitness implies the joint optimization of these metrics, since some can imply disturbances in the rest. The weight of each of them has been carefully tuned to find an optimal overall solution.

Except for the methods used in the paper, some of the most representative computational intelligence algorithms can be used to solve the problem, such as earthworm optimization algorithm (EWA) [40], monarch butterfly optimization (MBO) [41, 42], moth search (MS) algorithm [43], elephant herding optimization (EHO) [45], Harris hawks optimization (HHO) [46], and slime mould algorithm (SMA) [44].

6. Future Research

As possible future research lines, coefficients in equation (3) could be tuned using different algorithms of literature, analysing the optimization rate compared to the methods presented in this paper. The results of this paper have been tested in simulation; as future work, we propose to test the optimized controllers in real physical autonomous vehicles as well as to implement an online optimization using the best techniques studied in this paper.

Some dynamic metaheuristic optimization models, such as multilayer neural networks (MLP), could be implemented using the particular structure of MLPs. The MLP benefits from the complex architecture of the neural networks and its transformations in order to achieve new optimal solutions. In terms of convergence, the relationship between random exploitation and each parameter under asymmetric interval is derived and an iterative convergence of neural networks is proved mathematically [60]. Other optimization techniques such as the water cycle algorithm could also be applied to solve the problem proposed [61].

Abbreviations

ABC: Artificial bee colony algorithm
ALO: Ant lion optimizer
BA: Bat algorithm
BHO: Black hole optimization
CLONALG: Clonal selection algorithm
CS: Cuckoo search
CSO: Cat swarm optimization
DA: Dragonfly algorithm
DE: Differential evolution
EHO: Elephant herding optimization
EWA: Earthworm optimization algorithm
FFA: Firefly algorithm
GA: Genetic algorithms
GBS: Gravitational-based search
GOA: Grasshopper optimisation algorithm
GWO: Grey wolf optimizer
HHO: Harris hawks optimization
HS: Harmony search algorithm
KH: Krill herd algorithm
LiDAR: Light detection and ranging sensor
MBO: Monarch butterfly optimization
MFO: Moth flame optimizer
MS: Moth search algorithm
NNA: Neural network algorithm
NP: Nondeterministic polynomial-time
PID: Proportional integrative derivative
PSO: Particle swarm optimization
SCA: Sine cosine algorithm
SFL: Shuffled frog leaping
SMA: Slime mould algorithm
SSA: Salp swarm algorithm
WCA: Water cycle algorithm
WOA: Whale optimization algorithm.

Data Availability

No data is available.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

J.E.N.H. and N.G.B. conceptualized the study; L.F.d.M.L. and F.S. helped with software; J.E.N.H., F.S. and L.F.d.M.L. investigated the study; N.G.B. supervised; all authors analysed the data and wrote the paper. Luis Fernando de Mingo López, Francisco Serradilla, Jose Eugenio Naranjo Hernández, and Nuria Gómez Blas authors contributed equally to this work.

Acknowledgments

This work was partially supported by the Comunidad de Madrid under Convenio Plurianual with the Universidad Politécnica de Madrid in the actuation line of Programa de Excelencia para el Profesorado Universitario. This work has been partially supported by projects Seguridad de Vehículos AUTOmóviles para un TRansporte Inteligente, Eficiente y Seguro (SEGVAUTO4.0 P2018/EMT-4362) and CICYT (PID2019-104793RB-C33).

References

[1] S. Nesmachnow, "An overview of metaheuristics: accurate and efficient methods for optimisation," *International Journal of Metaheuristics*, vol. 3, p. 320, 2014.

[2] Z. Beheshti and S. M. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and its Applications*, vol. 5, pp. 1–35, 2013.

[3] O. P. Goldreich, *NP-Completeness: The Basics of Computational Complexity*, Cambridge University Press, Cambridge, MA, USA, 1st edition, 2010.

[4] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The Arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.

[5] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, Springer Publishing Company, New York, NY, USA, 1st edition, 2007.

[6] L. Nunes de Castro, "Fundamentals of natural computing: an overview," *Physics of Life Reviews*, vol. 4, pp. 1–36, 2007.

[7] Z. J. Lee, S. F. Su, C. C. Chuang, and K. H. Liu, "Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment," *Applied Soft Computing*, vol. 8, pp. 55–78, 2008.

[8] M. Dorigo and T. Stützle, *Ant Colony Optimization*, Bradford Company, Scituate, MA, USA, 2004.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford, UK, 1999.

[10] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 26, pp. 29–41, 1996.

[11] A. Askarzadeh, "Bird mating optimizer: an optimization algorithm inspired by bird mating strategies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, pp. 1213–1228, 2014.

[12] E. Duman, M. Uysal, and A. F. Alkaya, "Migrating birds optimization: a new metaheuristic approach and its performance on quadratic assignment problem," *Information Sciences*, vol. 217, pp. 65–77, 2012.

[13] A. Sharma, "A new optimizing algorithm using reincarnation concept," in *Proceedings of the 2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*, pp. 281–288, Budapest, Hungary, November 2010.

[14] H. Nguyen and B. Bhanu, "Zombie survival optimization: a swarm intelligence algorithm inspired by zombie foraging," in *Proceedings of the 2012 21st International Conference on Pattern Recognition (ICPR 2012)*, pp. 987–990, IEEE Computer Society, Los Alamitos, CA, USA, June 2012.

[15] D. Teodorovic, P. Lucic, G. Markovic, and M. D. Orco, "Bee colony optimization: principles and applications," in *Proceedings of the 2006 8th Seminar on Neural Network Applications in Electrical Engineering*, pp. 151–156, Serbia and Montenegro, Belgrade, September 2006.

[16] H. J. Sung, "Queen-bee evolution for genetic algorithms," *Electronics Letters*, vol. 39, pp. 575-576, 2003.

[17] H. A. Abbass, "MBO: marriage in honey bees optimization-a Haplometrosis polygynous swarming approach," *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 207–214, 2001.

[18] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings-IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[19] L. N. de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 239–251, 2002.

[20] S. Mirjalili and A. Lewis, "The Whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[21] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, pp. 129–154, 2006.

[22] T. Jayabarathi, S. Chalasani, Z. A. Shaik, and N. D. Kodali, "Hybrid differential evolution and particle swarm optimization based solutions to short term hydro thermal scheduling," *WSEAS Transactions on Power Systems*, vol. 11, pp. 245–254, 2007.

[23] M. Pant, R. Thangara, C. Grosan, and A. Abraham, "Hybrid differential evolution-particle swarm optimization algorithm for solving global optimization problems," in *Proceedings of the Third IEEE International Conference on Digital Information Management*, pp. 18–24, London, UK, November 2008.

[24] S. C. Chu, P. w. Tsai, and J. S. Pan, *Cat Swarm Optimization. PRICAI 2006: Trends in Artificial Intelligence*, Q. Yang and G. Webb, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[25] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[26] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Proceedings of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft Computing*, Cancun, Mexico, June 2007.

[27] X. S. Yang, *Firefly Algorithms for Multimodal Optimization. Stochastic Algorithms: Foundations and Applications*, O. Watanabe and T. Zeugmann, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2009.

[28] S. Mirjalili, S. M. Mirjalili, and A. G. W. Lewis Optimizer, *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[29] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, pp. 1053–1073, 2016.

[30] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.

[31] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, Cambridge, MA, USA, 1992.

[32] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, pp. 1567–1579, 2007.

[33] S. S. Mirjalili, "CA: a Sine Cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[34] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[35] A. H. Gandomi and A. H. K. Alavi herd, "A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 4831–4845, 2012.

[36] Y. Xin-She and H. G. Amir, "Bat algorithm: a novel approach for global engineering optimization," *Engineering Computations*, vol. 29, pp. 464–483, 2012.

[37] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, pp. 2232–2248, 2009.

[38] X. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the 2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, India, December 2009.

[39] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.

[40] G. Wang, S. Deb, and L. Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 12, pp. 1–22, 2018.

[41] H. Faris, I. Aljarah, and S. Mirjalili, "Improved Monarch butterfly optimization for unconstrained global search and neural network training," *Applied Intelligence*, vol. 48, pp. 445–464, 2018.

[42] Y. Feng, S. Deb, G. Wang, and A. H. Alavi, "Monarch butterfly optimization: a comprehensive review," *Expert Systems with Applications*, vol. 168, Article ID 114418, 2021.

[43] G. G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, pp. 151–164, 2016.

[44] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[45] G. Wang, S. Deb, and d. S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 1–5, Bali, Indonesia, December 2015.

[46] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[47] L. Abualigah and A. Diabat, "A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications," *Neural Computing and Applications*, vol. 32, pp. 15533–15556, 2020.

[48] L. Abualigah, "Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications," *Neural Computing and Applications*, vol. 33, pp. 2949–2972, 2020.

[49] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 220–222, 1993.

[50] K. Ogata, *Modern Control Engineering*, Prentice Hall PTR, Hoboken, NJ, USA, 4th edition, 2001.

[51] K. L. Chien, J. A. Hrons, and J. B. Reswick, "On the automatic control of generalized passive systems," *Transactions of the American Society of Mechanical Engineeing*, vol. 74, pp. 175–185, 1972.

[52] G. H. Cohen, "Theoretical consideration of retarded control," *Transactions of the American Society of Mechanical Engineeing*, vol. 75, pp. 827–834, 1953.

[53] L. Mora, R. Lugo, C. Moreno, and J. E. Amaya, "Parameters optimization of PID controllers using metaheuristics with physical implementation," in *Proceedings of the 2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–8, Valparaíso, Chile, October 2016.

[54] S. K. Valluru and M. Singh, "Optimization strategy of bio-inspired metaheuristic algorithms tuned PID controller for PMBDC actuated robotic manipulator," *Procedia Computer Science*, vol. 171, pp. 2040–2049, 2020.

[55] R. Matušů, K. Janprom, W. Permpoonsinsup, and S. Wangnipparnto, "Intelligent tuning of PID using metaheuristic optimization for temperature and relative humidity control of comfortable rooms," *Journal of Control Science and Engineering*, vol. 2020, Article ID 2596549, 2020.

[56] F. Serradilla, N. Cañas, and J. E. Naranjo, "Optimization of the energy consumption of electric motors through metaheuristics and PID controllers," *Electronics*, vol. 9, p. 1842, 2020.

[57] J. E. Naranjo, F. Serradilla, and F. Nashashibi, "Speed control optimization for autonomous vehicles with metaheuristics," *Electronics*, vol. 9, p. 551, 2020.

[58] J. Bhookya and R. K. Jatoth, "Optimal FOPID/PID controller parameters tuning for the AVR system based on sine-cosine-algorithm," *Evolutionary Intelligence*, vol. 12, pp. 725–733, 2019.

[59] S. Zhang and L. Liu, "Normalized robust FOPID controller regulation based on small gain theorem," *Complexity*, vol. 2018, Article ID 5690630, 2018.

[60] A. Sadollah, H. Sayyaadi, and A. Yadav, "A dynamic metaheuristic optimization model inspired by biological nervous systems: neural network algorithm," *Applied Soft Computing*, vol. 71, pp. 747–782, 2018.

[61] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm–a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers and Structures*, vol. 110-111, pp. 151–166, 2012.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4464890, 11 pages
https://doi.org/10.1155/2021/4464890

Research Article

Heterogeneous Signal Fusion Method in Driving Fatigue Detection Signals

Qingjun Wang [1,2] **and Zhendong Mu** [3]

[1] *Shenyang Aerospace University, Shenyang 110136, China*
[2] *Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[3] *The Center of Collaboration and Innovation, Jiangxi University of Technology, Nanchang 330098, China*

Correspondence should be addressed to Zhendong Mu; zdmu123@jxut.edu.cn

Received 15 July 2021; Revised 8 September 2021; Accepted 14 September 2021; Published 14 October 2021

Academic Editor: Chi-Hua Chen

Driving fatigue is a physiological phenomenon that often occurs during driving. When the driver enters a fatigue state, they will become distracted and unresponsive, which can easily lead to traffic accidents. The driving fatigue detection method based on a single information source has poor stability in a specific driving environment and has great limitations. This work helps with being able to judge the fatigue state of the driver more comprehensively and achieving a higher accuracy rate of driving fatigue detection. This work mainly introduces research into different signal fusion methods to detect fatigue drive. This work will take the normal driver's breathing signal, eye signals, and steering wheel signal as research objects and collect and isolate the characteristics of the fatigue detection signal. Research was then conducted on different signal fusion methods for the detected depth of breath. Change of steering angle, eyelid closure, and blinking marks and the fatigue driving experiment was designed to evaluate the results of different data fusion methods. Experimental results show that the detection accuracy of the heterogeneous signal fusion method in fatigue detection is as high as 80%.

1. Introduction

1.1. Research Background and Significance. With the continuous improvement of domestic infrastructure construction and the continuous improvement of people's economic income level, the number of private cars has expanded significantly in recent years. Vehicles, such as cars and trains, still rely on the manual driving of the driver, and driving fatigue is a physiological phenomenon that often occurs during driving and is unavoidable [1]. In highway traffic, fatigue driving has become one of the main causes of traffic accidents. According to the statistics, more than 60% of drivers have had a long continuous driving experience. Among the major traffic accidents in the entire country, 27% of the drivers are fatigued and cause the accident [2]. Toyota has conducted investigations into the causes of traffic accidents. Among the three key factors: people, cars, and roads [3], the traffic accidents directly or indirectly caused by the "people" factor accounted for 92.9%, and the vast majority of traffic accidents are directly or indirectly related to the status of the driver [4]. The driver's driving state directly affects the operating error rate and the ability to deal with emergencies. The fatigue that occurs during driving can cause the driver to become distracted and slow to react. Driving fatigue has become an invisible killer in road traffic and rail traffic. When the driver is tired and lethargic, the vehicle will not be able to control as expected. And the consequences are unpredictable and unexpected as well. This not only causes loss to one's own life and property, but also brings great harm to pedestrians. Therefore, it is urgent to solve the problem of fatigue driving. To prevent accidents such as fatigue driving, it is necessary to develop effective fatigue detection methods; at this stage, fatigue detection is carried out by a single signal, and the detection of a single signal is one-sided. Therefore, in fatigue detection, research on how to detect fusion signals is imperative, which can detect driver fatigue in multiple directions and reduce driver fatigue.

1.2. Related Content. Fatigue can affect normal work and even cause accidents. To reduce the impact of fatigue on people, Wang proposed a method to provide real-time fatigue detection [5]. First, he uses an active shape model to detect human faces and extracts the histogram features of directional gradients of the eyes and mouth. Secondly, the support vector machine (SVM) is used to classify the state and posture of the orthography and scaling with iterations algorithm to estimate the head posture. Third, according to the face state, the fatigue decision index is obtained, and the weight of the fatigue decision index is calculated by the entropy method. Finally, based on the calculated fatigue decision index, he applies the Bayesian method to evaluate the driver's fatigue level. The final average accuracy of his method is 83.3%. However, the operation of this method is more complicated. Li et al. explored a feature weight-driven signal fusion method [6] and proposed interactive mutual information modeling to improve the accuracy of mental workload classification. They used EEG and ECG signals to verify the proposed heterogeneous organisms the effectiveness of signal fusion methods. they invited ten subjects to participate in simple, medium, and difficult tasks to collect brain and ECG signals of different mental load levels. Then, they can be used for classification according to the heterogeneous physiological signals of different mental workload states. Their experiments show that the ECG can be used as a supplement to the EEG, optimize the fusion model, and improve the estimation of mental workload. The classification results show that the proposed biosignal fusion method IMIM can improve the classification accuracy of feature-level and classifier-level fusion. Their research shows that multimodal signal fusion is expected to identify the level of mental workload, and the fusion strategy has potential psychological workload estimation applications in cognitive activities in daily life. However, there are limitations of location in terms of use. The disadvantage is that only ten experimental subjects were selected, which is too small to have the validity of the experiment. Zheng et al. proposed a dynamic fatigue detection model based on hidden Markov model [7]. The model can use various physiological and detection information to estimate driver fatigue in a probabilistic manner. In the actual driving process, they simultaneously recorded electroencephalogram (EEG), electromyography (EMG), and breathing signals through wearable sensors and sent them to the computer via Bluetooth. Then, according to the physiological information, the distribution estimates of different time periods are used to obtain the fatigue probability. Their HMM-based fatigue identification method can dynamically obtain fatigue testing. However, the calculation amount of this detection method is too large, and the detection has greater difficulty. Pilataxi et al. introduced a driving assistance system that detects when the driver is drowsy [8]. The system is tested by a car-like robot that is wirelessly controlled by a computing interface developed in Visual Studio 2010, which simulates a car panel. An artificial vision system monitors the driver's head direction to determine whether the driver is drowsy. However, this study did not use heterogeneous signal fusion methods.

1.3. Main Content and Innovation. The main content of this article is to study the respiratory physiological signals, driver operation signals, and eye signal detection in the fatigue driving detection, collect and extract the characteristics of the physiological signals and eye detection signals, and then compare the signal fusion methods. The best decision-level fusion method is selected for the fusion of heterogeneous signals, and finally the high accuracy of the heterogeneous information fusion method in the detection of fatigue driving is confirmed by the detection of fatigue driving signals. The innovation of this paper is to combine the fatigue driving detection signal with heterogeneous signal fusion and achieve a high accuracy rate of fatigue driving detection by fusing the collected heterogeneous signals.

2. Fatigue Driving Concepts and Methods

2.1. Definition of Fatigue Driving. To allow readers to better understand the meaning of driving fatigue, I will talk about some theoretical knowledge as the theoretical basis for the article. Fatigue driving refers to the phenomenon that the driver has physical and psychological disorders and decreased driving skills after driving for a long time. Fatigue is a very complex physiological phenomenon, which can generally be divided into mental fatigue and physical fatigue. Mental fatigue is manifested as restlessness, loss of motivation, difficulty in concentration, slow thinking, low mood, decreased work efficiency, prolonged reaction time, and decreased work accuracy. Its continued development will cause headaches, dizziness, insomnia, and dysfunction of the cardiovascular system, respiratory system, and digestive system. Mental fatigue and physical fatigue often occur at the same time [9, 10]. The driving fatigue discussed in this article refers to the phenomenon of slow response and misjudgment when the driver is driving. Driving fatigue is often a mixture of physical fatigue and psychological fatigue [11]. Continuous driving time is the most important and common cause of driving fatigue. According to the statistics of the literature, after 3 hours of continuous driving, most drivers will start to feel fatigue and start to experience operational errors. After continuous driving, the incidence of car accidents will rise to 1.5 times that of normal conditions. Therefore, the laws of various countries have strictly controlled the driving time of drivers, as shown in Table 1.

The generation of driving fatigue is also closely related to the driver's physical and mental state when starting to drive. Factors such as the driver's circadian rhythm, sleep time, age, and gender will all induce driving fatigue to varying degrees. The driving environment is also a factor in driving fatigue. Conditions such as temperature, humidity, oxygen content in the air, noise, and vibration of car seats are much worse than indoor environments. Unfavorable conditions driving fatigue usually occurs in driving environments [12, 13].

2.2. Introduction to Over-Limit Learning Machine. Over-limit learning machine is an algorithm in neural network research. It is a generalized single hidden layer feedforward neural network used for classification,

Table 1: The control of driving time by laws of various countries.

Country	Continuous driving time	Cumulative driving time (one day)
China	4 hours	8 hours
United States	None	10 hours
Japan	4 hours	9 hours
Australia	5 hours	12 hours
Europe	4.5 hours	10 hours

regression, clustering, sparse approximation, compression, and single or multilayer hidden nodes. Feature learning [14, 15] is where the parameters of the hidden nodes (not just the weights connected to the input to the hidden nodes) do not need to be adjusted [16]. These hidden nodes can be randomly assigned and never updated (i.e., they are random projections, but have nonlinear transformations), or they can be inherited from their parents without being changed [17, 18]. In most cases, the output weights of hidden nodes are usually learned in a single step, which is equivalent to learning a linear model [19]. There will be no common problems such as falling into local minimum and overfitting in traditional neural networks, and it is suitable for modeling systems with complex nonlinear input and output relationships. Its advantage is that the learning speed is fast. For the algorithm applied to the product, it can greatly reduce the calculation amount, increase the calculation speed, and greatly improve the calculation speed [20, 21]. Its characteristic is that the weights of hidden layer nodes are randomly or artificially given and do not need to be updated. The learning process only calculates the output weights. Figure 1 shows the simple results and applications of over-limit learning.

The ELM structure is as follows: suppose there are K pairs of input and output combinations, x is the feature vector used for training, and y is the corresponding output result, where $i = 0$ [22, 23]. Assuming it is an N-dimensional feature vector, at the same time, the number of hidden nodes in the overlimit learning machine is also set to N [24]. Let U be a unitary matrix of size $N * N$ in the first layer of ELM. f is the nonlinear activation function of ELM, and W is the weight vector of the second layer of ELM. At this time, the output of ELM can be expressed as

$$[f(u)\Lambda f_{n-1}(u_i)]W = y. \tag{1}$$

2.3. Introduction to Fractional-Order System. Fractional-order system is a generalization of the integer order system. Many physical processes can be modeled by fractal sequence systems such as diffusion and thermal conductivity, diffusion circuits, electrochemical processes, polarized dielectric viscous materials such as polymers and rubber, the release phenomenon of organic dielectric materials, flexible structure data network and biological traffic, etc. [25, 26]. This article will be used in data modeling. The general fractional order system can be expressed as

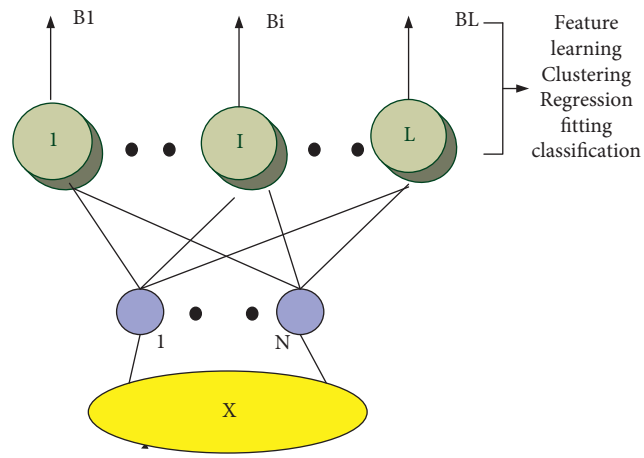


Figure 1: Simple structure and application of overlimit learning machine.

$$y(t) + \sum_{i=1}^{n} a y(t) = \sum_{i=1}^{m} b\, D(t). \tag{2}$$

According to the definition of Miller et al., the generalized form of the fractional difference is defined as

$$D_k^a f(k) = \lim \frac{1}{h} \sum_{m=0}^{a} (-1)^m \binom{a}{m}, \tag{3}$$

where h is the sampling period, where it is defined as

$$\binom{a}{m} = \frac{K(a+1)}{K(a-m+1)K(m+1)}, \tag{4}$$

where $K(a)$ is defined as

$$K(a) = \int_0^{+\infty} \varsigma^{z-1} e^{-\zeta} d. \tag{5}$$

In actual calculations, to simplify the calculations, matrix operators can be used to approximate and implement fractional differential operations; namely.

$$\begin{bmatrix} Df(h) \\ M \\ Df(kh) \end{bmatrix} = \frac{1}{h} \begin{bmatrix} w_0 & \Lambda & 0 \\ M & w_0 & 0 \\ w_k & \Lambda & w_0 \end{bmatrix}, \tag{6}$$

$$w_k = (-1)^k \binom{a}{k}.$$

Due to the edge effect and the truncation effect [27, 28], equation (6) cannot always start from $k = 1$. However, each row of the fractional differential operator matrix can be regarded as the weight of the input x. When $k > 1$ and a is not an integer, it will approach 0 [29, 30]. Therefore, a suitable upper limit K needs to be selected. At this time, formula (6) can be rewritten as

$$\begin{bmatrix} Df(kh) \\ M \\ Df(Th) \end{bmatrix} = \frac{1}{h^a} \begin{bmatrix} w_{k-1} & \Lambda & 0 \\ M & w_0 & 0 \\ 0 & \Lambda & w_0 \end{bmatrix}. \tag{7}$$

3. Feature Extraction and Heterogeneous Signal Fusion in Fatigue Detection

3.1. Physiological Signal Feature Collection

3.1.1. Working Principle of Detecting Physiological Signals. In this paper, continuous wave radar is selected as the detection sensor. The typical block diagram of continuous wave radar is shown in Figure 2, including power divider, transmitting antenna, receiving antenna, oscillator, mixer, detector, and other circuits. The transmitting antenna transmits the microwave signal in a directional direction, and it is reflected back after encountering obstacles and received by the receiving antenna. After being mixed with the oscillator, the low-frequency signal is detected by the mixer. When using continuous wave radar to detect the human thoracic cavity, the movement of the thoracic cavity will produce frequency modulation. According to the Doppler effect, the reflected radar signal will also have a corresponding frequency shift, so the obtained low-frequency signal contains the movement information of the thoracic cavity.

Human body thoracic cavity that is used as a target for detecting the radar emission signal returned by the thoracic cavity movement will generate frequency modulation; the phase information related to the thoracic cavity movement can retrieve the modulation data. Can be late, the phase data can reflect the tester's breath and heartbeat. Doppler radar has a good ability to penetrate clothing or bedding and can achieve noncontact detection. However, for the radio frequency, whether the signal is in the air or on the surface of the skin, there is reflection loss, which results in a strong correlation between the performance of the radar and the frequency. Therefore, the choice of radar sensor is a key indicator, which directly affects the quality of the experimental results. From the theory of continuous wave radar, it can be known that the smaller the wavelength of the radar in the output baseband signal $B(t)$, the more obvious the displacement change of the detected chest cavity, which means that the higher the frequency of the selected radar, the better. According to the electromagnetic field theory, the higher the radar frequency, the smaller the signal reflection after encountering the human body surface, the weaker the ability to penetrate clothing, and the greater the energy consumption of transmitting signals of the same power. It can be seen that the frequency selection of radar is a contradictory process. According to the reasons such as penetration ability, volume, and transmission power, in this article, the 10.525 GHz radar module HB100 is selected as the front-end detection sensor. This module is a microwave moving object detector designed by using the Doppler effect. The actual sample is shown in Figure 3. The radar module is mainly used for automatic door switches, safety precaution systems, automatic train signals, and other occasions. The internal integrated transmission and receiving antennas, oscillators, mixer, detectors, and other modules have the advantages of strong antiinterference ability, low output power, and long detection distance.

3.1.2. Physiological Signal-Related Quantity. Normal signals measured by Doppler radar include respiratory signals and heart rate signals. This article aims to study the relationship between the driver's breathing and the changes in heart rate while driving and the degree of fatigue. Feature value extraction of physiological information, where the breathing amplitude at a certain moment is represented by $H(t)$, which reflects the depth of respiration (RD), that is, the expiratory volume and inhalation volume during breathing, and the respiratory rate (RR) at a certain moment, is represented by $H(F)$ representation. The standard deviation can be used to measure the degree of deviation of a random variable from the mean. The standard deviation is

$$\begin{aligned} W1 &= \sqrt{\frac{1}{n}\sum_{t-1}^{t}\left(H(t) - Hb_{\text{ave}}\right)}, \\ W2 &= \sqrt{\frac{1}{n}\sum_{t=1}^{t}\left(H(t) - Hf_{\text{ave}}\right)}, \end{aligned} \tag{8}$$

where $W1$ and $W2$ represent the standard deviation of the respiratory amplitude and frequency, respectively, and Hb and Hf represent the average respiratory amplitude over a period of time. Because the heart rate amplitude of the radar test is very small, it cannot reflect the change of the heart rate amplitude. In this paper, the heartbeat signal frequency (HR) is used to test the driver's heart rate fluctuations during the failure phase, and the standard deviation $W3$ is also used to measure it, which can be expressed as

$$W3 = \sqrt{\frac{1}{n}\sum_{t=1}^{t}\left(Y_f(t) - Y_{\text{ave}}\right)}. \tag{9}$$

Among them, $Y(t)$ represents the heartbeat frequency at a certain moment and represents the average value of the heartbeat frequency over a period of time.

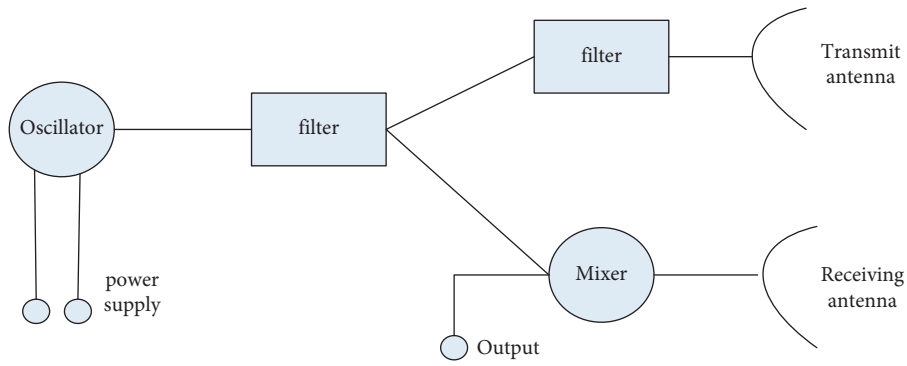


Figure 2: Block diagram of continuous wave radar.

Figure 3: The physical map of the radar module.

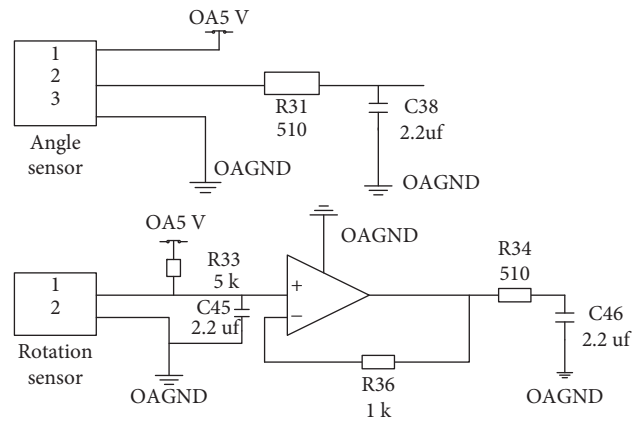


Figure 4: Schematic diagram of angle sensor.

3.2. Steering Wheel Angle Collection

3.2.1. Principle of Steering Wheel Acquisition Sensor. Most steering angles have two measurement methods: absolute and relative. This is because it is necessary to record the specific position of the steering wheel during the research process. An absolute angle encoder was therefore chosen as the sensor in this study. The angle of the steering wheel can be detected without affecting the driver, and the accuracy of the detection can be improved. According to the steering wheel steering requirements of fatigue driving, the model P3022 Hall angle sensor is selected. The P3022 Hall angle sensor converts the transformed angle signal into an electrical signal via electromagnetic induction principle; when the steering angle changes, the angle of the sensor installed under the steering wheel will change accordingly. And the sensor will output an electrical signal proportional to the rotation angle. P3022 angle sensor has the characteristics of 360-degree mechanical angle, angle resolution, 0.088 degrees, output 0–5 V analog voltage signal, and so on. Since the measurement range of the angle sensor is only 0–360 degrees, and the actual steering wheel rotation angle is 0–1080 degrees, which far exceeds the measurement range of the angle sensor, it is necessary to add a sensor to record the entire process of steering wheel movement. In this way, it is possible to accurately judge whether the driver is driving in a straight line and can also switch to the Hall angle sensor for timely measurements and accurately record small-turn signal changes. The circuit design is shown in Figure 4. The system uses a 3590S-2-103L precision ten-turn rotary potentiometer to measure the entire angle change during the rotation of the steering wheel.

The maximum resistance value of the rotary potentiometer is 5 KΩ. The actual change range of the potentiometer is 500 Ω-5 KΩ when the steering wheel rotates 0–1080 degrees. Considering the power consumption of the circuit and the signal change range, the voltage divider resistance is 5 KΩ, and the analog signal output change range is 0.45 V–2.5 V. It can be seen that the resolution of the rotary encoder is very low, but the experiment only needs to record the changing trend and position of the steering wheel angle, and there is no need to have high requirements for accuracy.

3.2.2. Reverse Disk Angle Signal Correlation Quantity. In this paper, the fatigue driving detection uses a rotary potentiometer to record the rotation track of the entire steering wheel. When the steering wheel rotates clockwise, the sampled voltage signal value will continue to increase, and when the steering wheel rotates counterclockwise, the sampled voltage signal value will continue to decrease. Therefore, it can be judged whether the current road condition is a continuous curve, a large curve, a small curve, or a straight road according to the change trend of the sampled voltage signal. In this study, the voltage value can be used directly instead of the angle value to calculate. Because the analysis is only concerned with the change trend of the steering wheel angle of the angle sensor, and the change of the small angle, the Hall angle sensor outputs the voltage value corresponding to the angle.

(1) The steering wheel angular velocity V can be expressed as

$$V(t) = \frac{\theta_i - \theta_{i-1}}{t_i - t_{i-1}}. \quad (10)$$

Among them, θ is the digital number of the output conversion of the voltage value corresponding to the steering angle of the steering wheel at t.

(2) The standard deviation of the steering wheel angular velocity can be expressed as

$$w_t = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(v_i - v_{\text{ave}})}. \quad (11)$$

Among them,

$$v_{\text{ave}} = \frac{1}{N}\sum_{i=1}^{n} v_i, \quad N \text{ is the number of samples.} \quad (12)$$

(3) The standard deviation of the steering angle of the steering wheel can be expressed as

$$w = \sqrt{\frac{1}{N-1}\sum_{i-1}^{n}(\theta_i - \theta_{\text{ave}})}. \quad (13)$$

Calculate the steering angular velocity, angular velocity standard deviation, and angle standard deviation per unit time during the driving process of the driver, and compare it with expert scoring standards to analyze the correlation between the degree of driving fatigue and the parameter information of the driving vehicle.

3.3. The Driver's Eye Feature Point Collection Based on the Over-Limit Learning Machine. Eye feature point positioning belongs to the category of face alignment, which refers to finding various facial feature points (landmarks) from the detected face, such as key positions such as eyes, eyebrows, nose, mouth, and facial contours. Figure 5 shows the facial feature point model.

As shown in Figure 6, the feature points are distributed in the lines in the figure, where the eye area contains the main information of the eyes, and the schematic diagram of the eye feature points is shown in Figure 6, including 2 points on the upper eyelid, 2 points on the lower eyelid, and 2 points on the left and right corners of the eye.

Over-limit learning machine is an algorithm in neural network research, used for classification, regression, clustering, sparse approximation, compression, and feature learning of single or multilayer hidden nodes. The algorithm builds a cascaded residual regression tree, thus making the face gradually return to the true position from the initial value of the feature point. Use the super learning machine to align the detected face area to find the position of the face. The detection result is shown in Figure 7. The eyelid aspect ratio can characterize the degree of eyelid opening. We select 3 typical eyelid closure types for analysis.

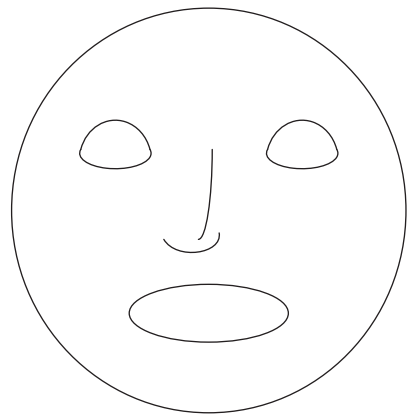


Figure 5: Location information of feature points.

For various eyelid states, the eyelid height-to-width ratio can be used to represent it. The eyelid height-to-width ratio can reflect the various transformation characteristics of the driver's eyes in a time series. The blink frequency refers to the number of blinks of the driver in a unit time. Under normal circumstances, the driver's blinking frequency remains within a relatively stable range during normal driving. When the driver is in a fatigued driving state, the blink frequency will be significantly reduced or significantly increased, which is very unstable. Eye-closing speed refers to the time it takes for the eyelids to go from a normal open state to a fully closed state. The shorter the time, the faster the eyelid closure speed, and the shorter the time, the slower the eyelid closure speed. The speed of closing the eyes of the driver during normal driving is much faster than during fatigue driving. When the driver is in a state of fatigue, the closing speed of the eyelids is very slow.

3.4. Heterogeneous Signal Fusion Method for Fatigue Driving Detection. Driving fatigue is a complex physiological and psychological phenomenon. The single-modal driving fatigue detection method cannot fully characterize the driver's fatigue. And in actual driving scenarios, due to various environmental interferences (such as unstable lighting conditions in the driving environment, vehicle vibration, spatial electromagnetic interference, etc.), the detection method based on a single information source is not reliable. This section discusses the combination of normal driver respiration data, eye data, and steering angle parameters and judge the overall fatigue condition of the driver, so as to achieve higher driving fatigue detection accuracy, stability, and environmental adaptability. Differential fusion means a complete preprocessing process. Data entry, forecasting, and arbitration decisions are based on information from many different sources to achieve more accurate, stable, and reliable target data from a single source.

3.4.1. Type of Signal Fusion. The heterogeneous signal fusion method actually uses the data of multiple sensors to obtain complete information about the object or the environment. Its core part is the fusion algorithm, and different fusion

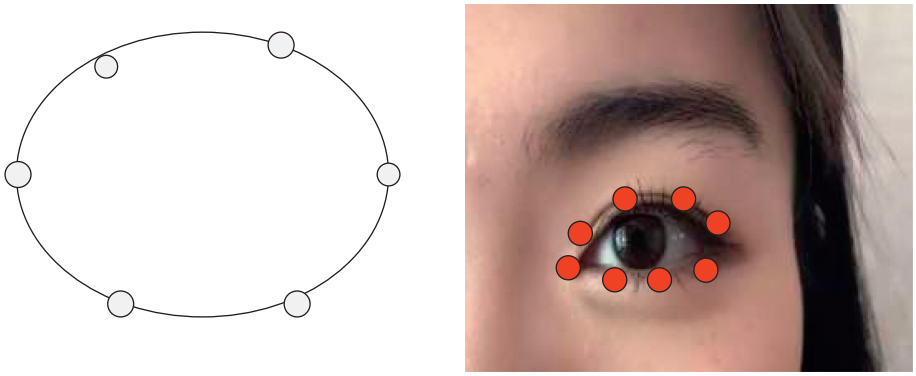


Figure 6: Eye feature point model and schematic diagram of eye feature point detection.

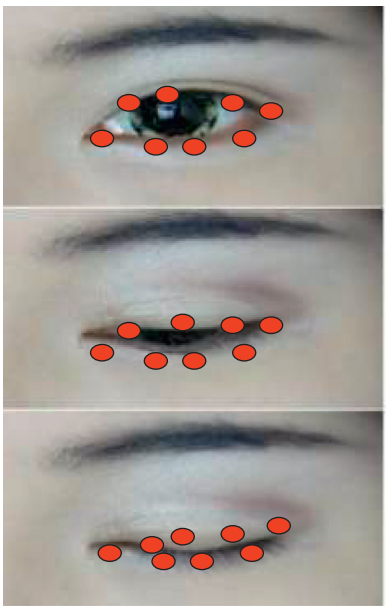


Figure 7: Typical degree of eyelid closure.

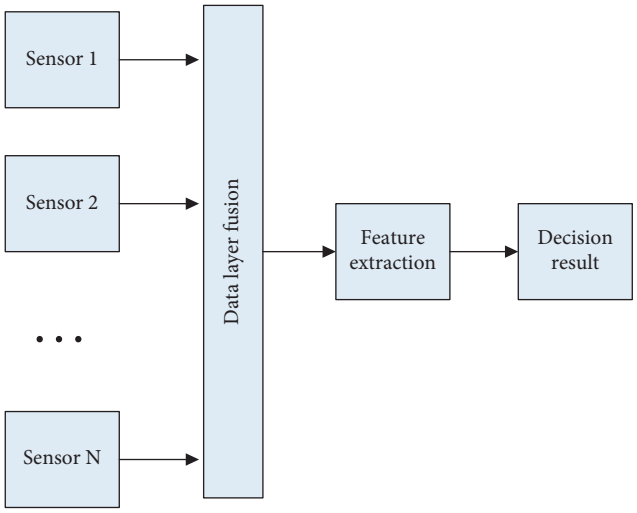


Figure 8: Data layer fusion.

algorithms have different advantages. There are many information fusion algorithms that have been proposed. According to the different levels of data abstraction, they can be divided into three levels: data layer fusion, feature layer fusion, and decision layer fusion. The integration of these three levels has its own advantages. As shown in Figure 8, the data layer fusion is to directly perform the fusion without any processing and processing of the raw data from the sensors of the same category, and then perform feature extraction with the data obtained after the fusion, and finally obtain the discrimination result. This method belongs to the lowest level of fusion, and there is no data loss. However, this fusion method has considerable limitations. It has certain restrictions on multiple sensors. The sensors to be fused must be of the same type, and the consistency of time and space must be ensured before fusion. This means that the increase in the amount of calculation will result in slow processing speed and poor real-time performance.

Compared with the other two methods, the feature-level fusion belongs to the middle-level fusion. Different from the data-level fusion, it first extracts the corresponding features from the original data collected by different sensors, performs certain processing on the extracted features, and then fuses them. Finally, the model is modeled to obtain the discrimination result. The fusion process is to fuse the features extracted from the original data, reducing a lot of calculations, improving the transmission speed and processing speed, and has a certain real-time performance, but compared with data-level fusion, a certain amount of information will still be lost. This leads to a decrease in accuracy. Feature layer fusion is shown in Figure 9.

Decision-level fusion is the highest level method among the three fusion methods. Unlike the previous two, it uses multiple sensors to monitor an object at the same time. However, unlike the data level, different types of sensors can be used, and each sensor is processed independently. That is, the model is established after the features are extracted, and the decision result is obtained, and then all decision results are merged and judged to obtain the final decision result. The decision fusion method is aimed at the decision results of different sensors, which makes it more computationally intensive and more real-time. However, it has undergone multiple layers of information extraction, and the information loss is more serious, which greatly reduces the accuracy of the fusion result. Therefore, the heterogeneous

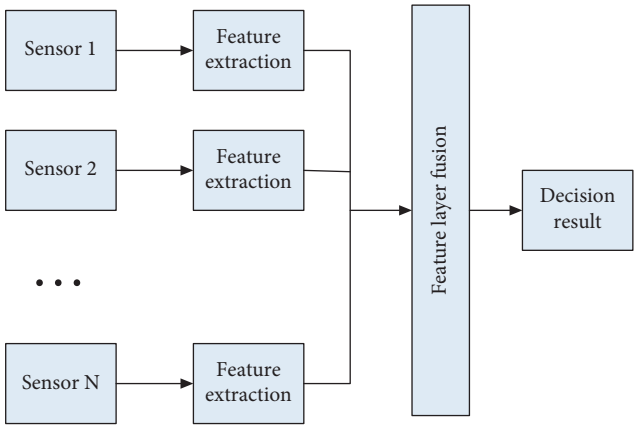


Figure 9: Feature layer fusion.

signal fusion in this paper adopts the decision-making layer fusion calculation, and the decision-making layer fusion is shown in Figure 10.

3.4.2. Decision-Making Layer Fusion of Heterogeneous Signals in Fatigue Driving Detection. The core idea of applying the information fusion method to fatigue driving detection is to analyze data from multiple angles to obtain the final high-precision identification result. The steps of decision-making and fusion of heterogeneous signals in the fatigue driving detection in this part are shown in Figure 11.

(1) Collection of characteristic data: Experiments are conducted in a good city road environment. At the same time, the above three devices collect data under different driving behaviors and simply process the collected sensor raw data: cutting samples, labeling, and the obtained data as the information converged data layer.

(2) Feature extraction: The feature extraction is performed on the data of the three information sources. The feature extraction method is the same as the above.

The same method is adopted, and the feature is obtained as the feature layer of information fusion.

(3) Train their respective identification models. Train the three features of the feature layer using different overlimit learning machine algorithms.

Three identification models are obtained, and their respective decision-making results are obtained.

(4) Integration of decision-making levels: The decision results obtained in the third step are fused through the designed fusion method to obtain the final recognition result.

4. Experiments and Results of Heterogeneous Information Fusion Methods

Based on the feature collection of EOG, ECG, and steering wheel proposed above and the extraction algorithm of several features such as heart rate, blink frequency, eyelid closure, and the final decision fusion algorithm, this article

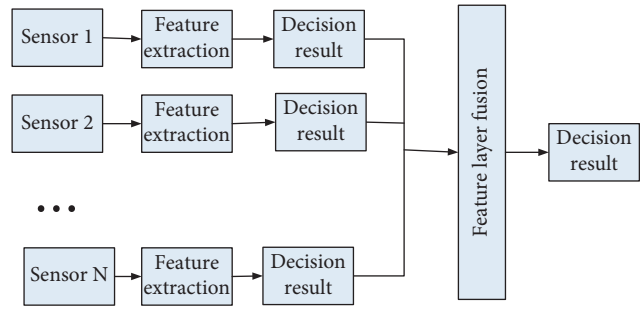


Figure 10: Decision-making layer fusion.

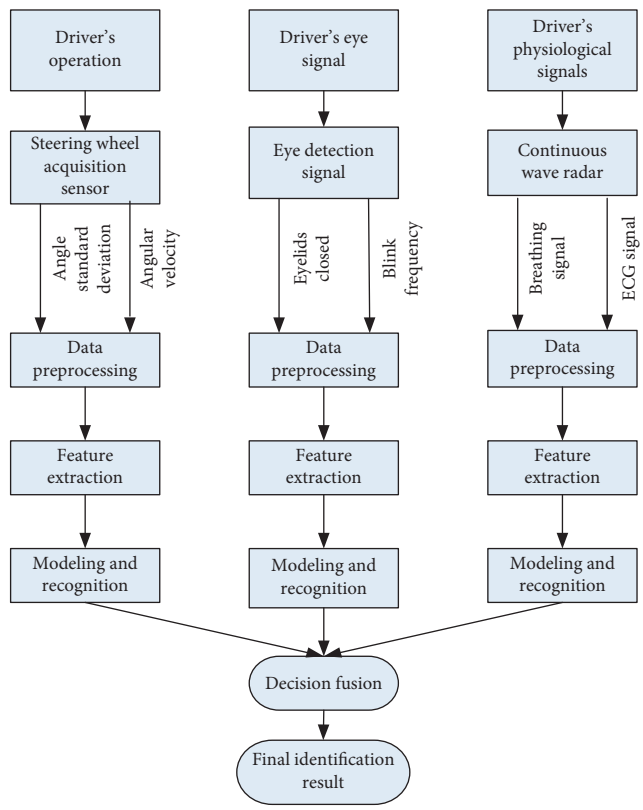


Figure 11: Decision-making layer fusion steps.

designs a set of fatigue verification methods to verify the feasibility of the fatigue detection methods designed in this article. In the article, a certain fatigue-induced paradigm was selected and the physiological data of 10 subjects were collected, and then the results were statistically verified.

4.1. Fatigue Induction Method. The main purpose of the experiment here is to induce fatigue in the subjects, measure its various physiological indexes under the state of fatigue occurrence, and make calculations and analysis. The obtained characteristic index is compared with the index calculated from the collected signals in the mental state, and finally the judgment result of the fatigue state is obtained. This experiment requires a more natural and undisturbed induction method. Choosing the method of neutral long-term video stimulation can make the individual produce a

spontaneous fatigue response. In the actual experiment arrangement, this article selected 5 subjects who are adult, healthy, non-drug-addicted individuals who are not taking any drugs, and the subjects are all young people aged 20–35 to ensure that the data is comparable. Five subjects were subjected to 25 minutes of real-time data collection of ECG, breathing rate, eye movement, and steering wheel information. A certain neutral video stimulus was given during the acquisition process to induce sleepiness in the subjects, and the physiological parameters in the process were recorded, and the characteristic indexes were calculated. The subjects were still awake at the beginning of the experiment. The parameter characteristics are compared, and the final result is obtained.

4.2. Data Analysis. Using the algorithm proposed in Section 2, this paper calculates the collected data and obtains its characteristic parameters by statistics. The detailed results are shown in Figure 12.

Statistical recognition rates of fatigue driving samples were 81%, 75%, 79%, 76%, and 76%, respectively. Judging by the predictions of the Extreme Learning Machine, the detection rate of the alarm state was as high as 81% of the steering wheel signal under varying degrees of fatigue in the steering wheel change trend. The signal currently predicts the level of fatigue while driving. In drowsy conditions, the range of steering angle changes is gradually reduced and the number of settings is significantly reduced. The depth of breathing in the normal signals of the driver is accompanied by a deeper sleep. The depth and width of the breath and the frequency gradually decrease.

Figure 13 shows the blink frequency per unit time calculated from the electrooculogram signal. This article chooses to use 1 minute as the unit. It can be clearly seen that as the experiment progresses, the blink frequency of the five subjects has increased significantly.

Based on the above statistical results, it can be seen that there is a certain correlation between blinking frequency and fatigue. The higher the blinking frequency, the more fatigued. Participants blinked more frequently when they entered a state of fatigue. In addition, it can be seen from Figure 13 that as fatigue deepens, the blink frequency of many subjects has a certain downward trend relative to the maximum value.

Figure 14 shows the eyelid distance per unit time calculated from the eye movement signal. The unit of 1 min is chosen. It can be clearly seen that as the experiment progresses, the eyelid distance gradually decreases in all five subjects.

From the above figure, it appears that the degree of eyelid closure can be used as a relative indicator to determine fatigue. In addition, Figure 14 shows that the fourth person sleeps in 15 minutes, when their eyelids are closed, and woke up after 16 minutes. Although different from the data of other volunteers, the general trend of change for each person, of course, is a downward one.

In summary, the experimental results show that the fatigue detection method of heterogeneous signal fusion can improve the accuracy of fatigue detection while avoiding the impact on driver behavior.

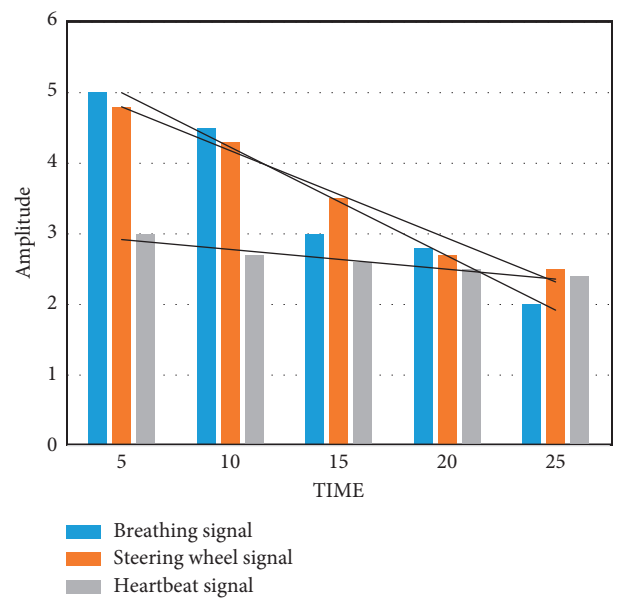


Figure 12: Physiological signal and steering wheel signal change amplitude.

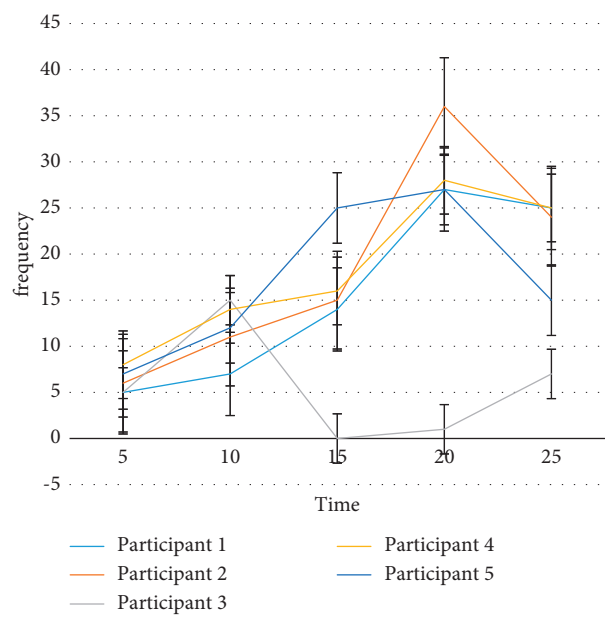


Figure 13: A statistical graph of the blink frequency of five subjects.

5. Discussion

This paper firstly collects and extracts features from physiological signals, driver operation signals, and eye signals in fatigue detection, and then performs signal fusion on the extracted signals to achieve a higher accuracy rate of fatigue driving detection, and finally performs signal fusion. In the detection experiment of the method, it is concluded that the accuracy of the fatigue driving detection method of the

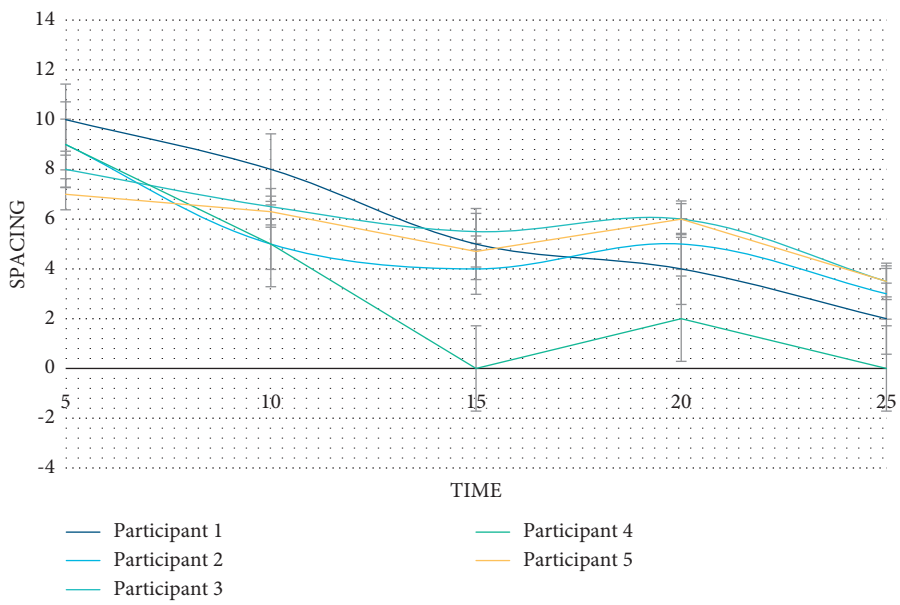


Figure 14: Statistics of eyelid closure data of five recipients.

heterogeneous signal fusion method is higher than that of the driving fatigue detection method of a single signal source through the statistics of the super learning machine algorithm. The research of this article has not enough understanding of the signal fusion method, but the research of this article has certain reference value for the detection method of fatigue driving. It provides a certain research route for future fatigue driving detection methods. This article still has some shortcomings. For example, the number of samples selected in our research process is not very large, and the data obtained is often not comprehensive enough. Driving fatigue is a problem that drivers often face. Combining with heterogeneous signal fusion methods can certainly make the research on driving fatigue more in-depth.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this research article.

Acknowledgments

This work was supported by the Project of Department of Education, Jiangxi Province (no. GJJ191000).

References

[1] A. Němcová, Y. Tian, and H. Jia, "Driving fatigue detection based on feature fusion of information entropy," *Journal of Computational Methods in Science and Engineering*, vol. 18, no. 8, pp. 1–12, 2018.

[2] Z. Liu, W. Zhang, S. Lin, and T. Q. S. Quek, "Heterogeneous sensor data fusion by deep multimodal encoding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 479–491, 2017.

[3] R. Fu, H. Wang, and W. Zhao, "Dynamic driver fatigue detection using hidden Markov model in real driving condition," *Expert Systems with Applications*, vol. 63, no. 5, pp. 397–411, 2016.

[4] L. Wang, C. Zhang, X. Yin, R. Fu, and H. Wang, "A non-contact driving fatigue detection technique based on driver's physiological signals," *Qiche Gongcheng/Automotive Engineering*, vol. 40, no. 3, pp. 333–341, 2018.

[5] A. Němcová, O. Janoušek, M. Vítek, and I. Provazník, "Testing of features for fatigue detection inEOG," *Bio-Medical Materials and Engineering*, vol. 28, no. 4, pp. 379–392, 2017.

[6] G. Li, B. Li, G. Wang, J. Zhang, and J. Wang, "A new method for human mental fatigue detection with several EEG channels," *Journal of Medical and Biological Engineering*, vol. 37, no. 2, pp. 240–247, 2017.

[7] C. Zheng, B. Xiaojuan, and W. Yu, "Fatigue driving detection based on Haar feature and extreme learning machine," *The Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 4, pp. 91–100, 2016.

[8] J. Pilataxi, W. Vinan, and D. Chavez, "Design and implementation of a driving assistance system in a car-like robot when fatigue in the user is detected," *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 457–462, 2016.

[9] M. Kolodziej, P. Tarnowski, D. J. Sawicki et al., "Fatigue detection caused by office work with the use of EOG signal," *IEEE Sensors Journal*, vol. 20, no. 24, p. 1, 2020.

[10] J. Yin, J. Hu, and Z. Mu, "Developing and evaluating a mobile driver fatigue detection network based on electroencephalograph signals," *Healthcare Technology Letters*, vol. 4, no. 1, pp. 34–38, 2017.

[11] F. You, Y. Gong, Y. Gong, X. Li, and H. Wang, "R2DS: a novel hierarchical framework for driver fatigue detection in mountain freeway," *Mathematical Biosciences and Engineering*, vol. 17, no. 4, pp. 3356–3381, 2020.

[12] Z. K. Gao, Y. L. Li, Y. X. Yang, and C. Ma, "A recurrence network-based convolutional neural network for fatigue

driving detection from EEG," *Chaos*, vol. 29, no. 11, Article ID 113126, 2019.

[13] L. G. Hernandez-Rojas, E. Martinez, and J. M. Antelis, "Detection of emergency braking intention using driver's electroencephalographic signals," *IEEE Latin America Transactions*, vol. 17, no. 01, pp. 111–118, 2019.

[14] M. Hanamura, T. Sawada, and T. Serizawa, "In-paper self-assembly of cellulose oligomers for the preparation of all-cellulose functional paper," *ACS Sustainable Chemistry & Engineering*, vol. 9, no. 16, pp. 5684–5692, 2021.

[15] Y. Zhu, J. Li, X. Lin, X. Huang, and M. R. Hoffmann, "Single-cell phenotypic analysis and digital molecular detection linkable by a hydrogel bead-based platform," *ACS Applied Bio Materials*, vol. 4, no. 3, pp. 2664–2674, 2021.

[16] H. Sun, S. P. Chen, and L. P. Xu, "Research on cloud computing modeling based on fusion difference method and self-adaptive threshold segmentation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 6, pp. 1859010.1–1859010.15, 2018.

[17] W. Yan, J. Tan, H. Zhan, and H. Wang, "Research on the method of fault diagnosis based on multiple classifiers fusion," *International Journal of Hospitality Information Technology*, vol. 9, no. 2, pp. 195–202, 2016.

[18] X. J. Qin, Z. J. Duan, H. B. Zheng, and Y. Tang, "Efficient smoothness-preserving fusion modelling method for mesh models," *International Journal of Simulation Modelling*, vol. 16, no. 3, pp. 527–540, 2017.

[19] C. Ren, Y. J. Liang, X. J. Lu, and H. B. Yan, "Research on the soil moisture sliding estimation method using the LS-SVM based on multi-satellite fusion," *International Journal of Remote Sensing*, vol. 40, no. 5-6, pp. 2104–2119, 2019.

[20] S. Hai-jiao, L. Wei-ning, W. Jia-cheng, L. Pei-xun, L. Zhi-gen, and D. Yao-hua, "Data fusion method in multi-sensors autonomous tracking," *Chinese Journal of Liquid Crystals and Displays*, vol. 31, no. 8, pp. 801–809, 2016.

[21] L. Mingyu, C. Chi, C. Ching-Hsiang, and L. Wing, "A Gaussian process data modelling and maximum likelihood data fusion method for multi-sensor CMM measurement of freeform surfaces," *Applied Sciences*, vol. 6, no. 12, p. 409, 2016.

[22] C. Li and X. Yang, "Multifocus image fusion method using discrete fractional wavelet transform and improved fusion rules," *Journal of Modern Optics*, vol. 68, no. 11, pp. 1–13, 2021.

[23] X. S. Yang and S. W. Lee, "A study on clothing pattern optimization fusion method using 3D program," *The Korean Society of Science & Art*, vol. 38, no. 5, pp. 317–327, 2020.

[24] X. Jing, S. Li, J. Cheng, and G. Junjun, "Multidimensional situational information fusion method for energy saving on campus," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 4, pp. 1–15, 2020.

[25] Q. Chen, Z. Wang, and Y. Chai, "Multi-focus image fusion method based on improved VGG network," *Journal of Applied Optics*, vol. 41, no. 3, pp. 500–507, 2020.

[26] E. A. Veshkin, V. I. Postnov, V. V. Semenychev, and A. A. Barannikov, "Study of the properties of carbon plastic samples formed by the infusion method," *Industrial Laboratory. Diagnostics of Materials*, vol. 86, no. 3, pp. 39–43, 2020.

[27] G. Ravikanth, K. Sunitha, and B. Eswara Reddy, "Location related signals with satellite image fusion method using visual image integration method," *Computer Systems Science and Engineering*, vol. 35, no. 5, pp. 385–393, 2020.

[28] H. Park, "An RGB-NIR image fusion method for improving feature matching," *International Journal of Engineering and Technology Innovation*, vol. 10, no. 3, pp. 225–234, 2020.

[29] J. Cheng, C. Cai, X. Tang, V. S. Sheng, W. Guo, and M. Li, "A DDoS attack information fusion method based on CNN for multi-element data," *Computers, Materials & Continua*, vol. 62, no. 3, pp. 131–150, 2020.

[30] T. Smets, T. De Keyser, T. Tousseyn, E. Waelkens, and B. De Moor, "Correspondence-aware manifold learning for microscopic and spatial omics imaging: a novel data fusion method bringing mass spectrometry imaging to a cellular resolution," *Analytical Chemistry*, vol. 93, no. 7, pp. 3452–3460, 2021.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5815280, 15 pages
https://doi.org/10.1155/2021/5815280

Research Article

Short-Term Traffic Prediction considering Spatial-Temporal Characteristics of Freeway Flow

Jiaqi Wang, Yingying Ma, Xianling Yang, Teng Li, and Haoxi Wei

Department of Transportation Engineering, South China University of Technology, 381 Wushan Road, Guangzhou 510641, China

Correspondence should be addressed to Yingying Ma; mayingying@scut.edu.cn

Received 16 April 2021; Revised 15 July 2021; Accepted 7 September 2021; Published 13 October 2021

Academic Editor: Feng-Jang Hwang

This paper presents a short-term traffic prediction method, which takes the historical data of upstream points and prediction point itself and their spatial-temporal characteristics into consideration. First, the Gaussian mixture model (GMM) based on Kullback–Leibler divergence and Grey relation analysis coefficient calculated by the data in the corresponding period is proposed. It can select upstream points that have a great impact on prediction point to reduce computation and increase accuracy in the next prediction work. Second, the hybrid model constructed by long short-term memory and K-nearest neighbor (LSTM-KNN) algorithm using transformed grey wolf optimization is discussed. Parallel computing is used in this part to reduce complexity. Third, some meaningful experiments are carried out using real data with different upstream points, time steps, and prediction model structures. The results show that GMM can improve the accuracy of the multifactor models, such as the support vector machines, the KNN, and the multi-LSTM. Compared with other conventional models, the TGWO-LSTM-KNN prediction model has better accuracy and stability. Since the proposed method is able to export the prediction dataset of upstream and prediction points simultaneously, it can be applied to collaborative management and also has good potential prospects for application in freeway networks.

1. Introduction

Intelligent transportation system (ITS) has become an effective way to reduce pollution and improves the performance of freeways, while the short-term traffic flow prediction is an important part to support the smart management and control of freeways. The trend of short-term traffic flow prediction is changing from parametric statistical models to nonparametric models and mixed models. Time-series methods were widely used in parametric statistical models, including exponential smoothing [1–3], moving average [4, 5], and autoregressive integrated moving average (ARIMA) model [6–8]. Kalman filtering was also used for traffic flow prediction, such as adaptive Kalman filter [9–11], hybrid dual Kalman filter [12], and noise-identified Kalman filter [13]. With the rapid development of ITS and improvement of data quality, more nonparametric prediction methods are used in the prediction of traffic flow. K-Nearest Neighbor (KNN) nonparametric regression, a nonlinear prediction method, was used to calculate Euclidean distance to find the nearest neighbor for prediction [14]. The improved Bayesian combination model was proposed to increase the accuracy of prediction [15]. Support vector machines (SVM) were also used considering the weak sensitivity to outliers [16]. The combined algorithm based on wavelet packet analysis and least square support vector machines was used to resolve the uncertainty and randomness of data [17]. Particle swarm optimization (PSO) and other optimization algorithms were applied to SVM because of small model calculation and good prediction performance [18]. With the development of artificial intelligence (AI), deep learning models have been widely used in traffic prediction. Smith and Demetsky [19] used back-propagation (BP) neural network to do the prediction. Optimization algorithms such as PSO and genetic algorithm (GA) were also applied to BP, and the effect is obvious [20, 21]. Recurrent neural network (RNN) can realize long-term memory calculation and was used in prediction, but it

had the problem of gradient explosion [22]. Long short-term memory (LSTM) network was proposed to solve it by using a forget gate [23, 24], which was not only used in natural language processing [25], for example, language generation [26], text classification [27], and phoneme classification [28], but also in prediction fields, such as short-term traffic flow prediction [29], housing load prediction [30], and pedestrian trajectory prediction [31]. Furthermore, improvements and combinations with other models have been proposed in many fields, from application in large-scale data problems [32] to the prediction of traffic flow, such as using GA to optimize the LSTM hyperparameters to get better performance [33]. The comparison of typical machine learning models is shown in Table 1.

Deep learning models are widely used in traffic flow prediction, especially in short-term prediction [41]. However, traffic flow has strong spatial-temporal characteristics on time series [42, 43]. More attention was paid to this characteristic in recent years' research of short-term traffic flow prediction [44–46]. Luo et al. [40] proposed a spatial-temporal traffic flow prediction model with KNN and LSTM to screen highly correlated upstream points and produced the prediction. Ma et al. put forward a method to select input data for daily traffic flow forecasting through contextual mining and intraday pattern recognition [47] and produced the daily traffic flow forecasting with CNN and LSTM [48]. Supervisory learning was used to mine the relationship between the factors of historical data and current traffic flow to train the predictor in advance so as to reduce the predicting time [49]. In addition, the match-then-predict method [50] and the fuzzy hybrid framework [51] with dynamic weights by mining spatial-temporal correlations were both proposed. Attention mechanisms were also combined in LSTM to increase the accuracy of prediction [52]. These methods that combine various factors using attention mechanism can reasonably allocate limited resources, increase the efficiency, and reduce computation.

In this paper, we propose the short-term traffic flow prediction model considering the spatial-temporal characteristics using LSTM and KNN under the concept of attention mechanism. First, the Gaussian mixture model (GMM) is used to select the upstream detection points to produce the prediction. Two parameters are used for the classification: one is the Kullback–Leibler Divergence (KL), also known as the relative entropy, which reflects the difference in the distribution of two datasets through approximate calculations, especially for large-sample traffic data. The other is the grey relation analysis (GRA) coefficient, which reflects the correlation between two groups of normalized data after similarity analysis. Second, the hybrid model of LSTM and KNN is proposed to produce the prediction using the selected data. LSTM is used to predict the traffic flow of upstream points as the training dataset of KNN. To solve the problem of time lag, input time of upstream data is changed in the model according to the space distance between the input point and the prediction point and the average speed of traffic flow. Moreover, transformed grey wolf optimizer (TGWO) is used to optimize key parameters, and Savitzky-Golay (SG) filter smoothing is used to reduce the noise in the model to improve the performance. The proposed TGWO-LSTM-KNN prediction model in this paper gives greater consideration to the spatial-temporal characteristic of freeway traffic flow to improve the accuracy of prediction and reduces the complexity of computation by selecting and preprocessing input data.

The rest of this paper is organized as follows. Section 2 introduces the methodology of the proposed model. Section 3 carries out the experiments and analysis of the proposed model with real-world traffic flow data. Section 4 presents the conclusions and the prospects of the research. The abbreviations used in the rest of the paper are listed in Table 2.

2. Methodology

2.1. Framework. This paper proposes TGWO-LSTM-KNN with the GMM classification model, which includes two parts: data preparation and prediction. GMM is used to choose the input data in the data preparation part considering the spatial-temporal characteristics of freeway traffic flow, while the prediction part is composed of LSTM parallel computing module, KNN module, and TGWO module. The framework of the proposed model is shown in Figure 1.

2.1.1. Data Preparation. In the freeway network, the traffic flow of upstream correlates with the prediction point flow, which is considered as spatial correlation. Moreover, the traffic flow of the prediction point changes over time inter- and intraday, and each specific period may have different patterns, such as the morning peak hour and the off-peak hour. Therefore, time series are divided into different parts according to the flow patterns, which will help to improve the accuracy of prediction. The temporal characteristic of prediction point flow is observed through a flow chart. Then set the midpoint of two adjacent extreme points and complete the time series division task.

Related upstream sections are analyzed and selected using GMM binary classification. In this paper, two parameters are used as the classification criteria. One is the KL divergence, which is a commonly used method in information science to quantify the difference between two datasets. In a large-sample traffic dataset with complex distribution, the difference can be simply and quickly reflected. The other is the GRA coefficient, which can analyze the linear similarity between two datasets through a small amount of data. These two parameters can well reflect the correlation of upstream sections and predicted sections. The steps of the classification part are as follows:

Step 1: use time step and speed to determine the space range. Note the upstream points within the scope as $O_1 \sim O_m$.

Step 2: divide day time into $T_1 \sim T_\eta$.

Step 3: construct dataset. Divide the dataset into working days and nonworking days. Change the input time of upstream data to meet the time lag of the prediction point considering the distance and travel speed.

Table 1: Comparison of machine learning models in short-term traffic flow prediction.

The basic model	Model performance	Improved models
Backpropagation neural network (BP)	Advantages [34] (1) Self-adaptive self-learning (2) Strong nonlinear mapping ability (3) Strong generalization ability Disadvantages [34] (1) Slow convergence speed down (2) Easy to fall into the local optimal solution	GA-BP [21], PSO-BP [20], EEMD-IAGA-BP [35]
Least-squares support vector machines (LSSVM)	Advantages [36] (1) Suitable for small datasets (2) Simple and convenient (3) Strong nonlinear generalization ability Disadvantages [36] (1) Unable to handle large datasets (2) Sensitive to missing values	W-LSSVM [17], PSO-SVM [18], EMD-GPSO-SVM [37]
Long short-term memory (LSTM)	Advantages [38] (1) High precision (2) Solve the problem of RNN gradient Disadvantages [38] (1) Time-consuming calculation (2) Large consumption of hardware resources (3) Too many parameters, easy to overfit	GA-LSTM [33], SVD-PSO-LSTM [39], KNN-LSTM [40]

Table 2: Abbreviation table for algorithm required in this paper.

Name	Abbreviation
Long short-term memory	LSTM
K-nearest neighbor algorithm	KNN
Kullback–Leibler	KL
Grey relation analysis	GRA
Gaussian mixture model	GMM
Expectation maximization	EM
Transformed grey wolf optimization	TGWO
Support vector machines	SVM
Backpropagation neural network	BP
Root mean square error	RMSE
Mean absolute error	MAE
Mean absolute percentage error	MAPE

Step 4: calculate the KL divergence and the GRA coefficient in T_i of working days and nonworking days.

Step 5: input the KL divergence and the GRA coefficient into GMM for binary classification. $O_1 \sim O_m$ are divided into two groups in each T_i. The group of points with the KL divergence close to 0 and the GRA coefficient close to 1 is used as the strongly related section of the prediction point for the next prediction.

2.1.2. Prediction. The prediction part consists of three modules, which are LSTM module, KNN module, and TGWO module. KNN is selected at the bottom of the model considering the spatial features of freeway flow with the advantages of fast calculation speed and no lag. LSTM is used to predict the short-term traffic flow of upstream sections and then put the prediction results of upstream sections into KNN to predict the prediction point traffic flow. Because the relationship among upstream points is ignored in the model, multithread LSTM parallel computing (LSTMs) is used to reduce the time consumption of prediction. Also, to improve the performance of LSTM-KNN, TGWO is used to optimize the parameters of LSTM and KNN.

Step 1: use TGWO to optimize the steps and epochs in LSTM, K value in KNN.

Step 2: multithread LSTM parallel computing is used to reduce calculation by ignoring the relationship among upstream points. Each O_i is input into the corresponding LSTM module and then the output P_i set and D_0 together form a new dataset.

Step 3: input the dataset into the KNN module to predict the traffic flow and output D_0^p.

2.2. Data Preparation. There are three steps in data preparation: determination of spatial scope, time-series division, and GMM classification.

2.2.1. Determination of Spatial Scope. Since the time step of short-term traffic flow prediction (T_{step}) is usually less than one hour and the highway speed (V) is limited, the radius range of spatial scope can be calculated:

$$R = T_{\text{step}} \times V. \tag{1}$$

The accesses and ramps within the radius of the prediction point centered are selected as upstream points.

2.2.2. Time Series Division. There are random fluctuations in daytime traffic flow, as shown in Figure 2. SG is a method to smooth data based on local least-squares polynomial approximation proposed by Savitzky and Golay [53], which is used to get the extreme traffic value points of the prediction point (D_0) (see Figure 2). Set the point in the middle of the

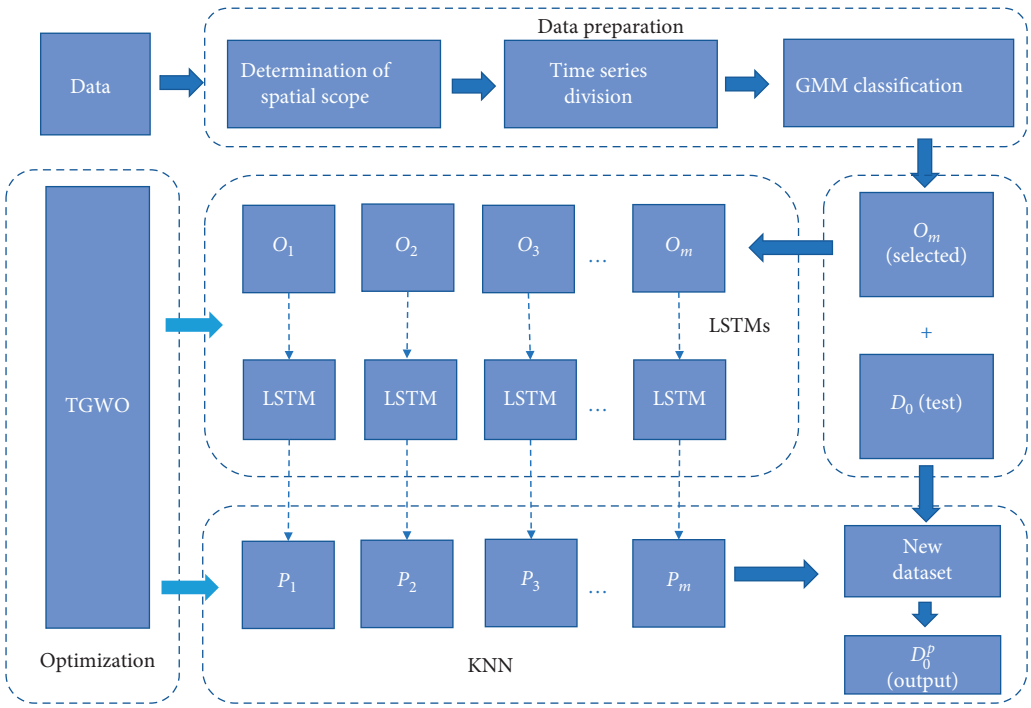


Figure 1: Framework of TGWO-LSTM-KNN with GMM classification.

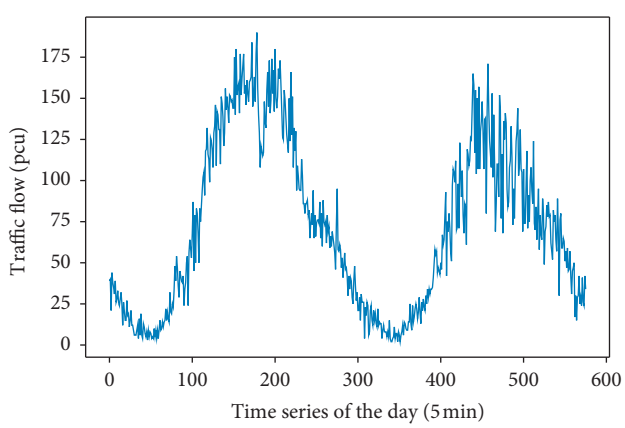


Figure 2: Traffic flow of the day before SG.

two adjacent extreme values, and each time series between two middle points is a time part (T_i) (see Figure 3).

2.2.3. SG Calculation Method. SG includes two parameters, which are the window length n and the order number k. For the window length n, with the increasing of n, the deviation between the processed data and the real data increases, and also the smoothness. For order number k, with the increasing of k, the deviation between the processed data and the real data decreases, and also the smoothness. According to the characteristics of highway traffic flow and existing research, the choice of n and k in this paper is 31 and 1, respectively.

Input upstream points data $O_m = (o_m(1), o_m(2), \ldots, o_m(x))$, where x denotes the length of data. Select the window length n and order number k. The data in a window are $O_m^n = (o_m(1), o_m(2), \ldots, o_m(n))$. The fitting polynomial is obtained using the $k-1$ least square method as follows:

$$o_m^{k-1}(1) = a_0 + a_1 o_m(1) + a_2 o_m(1)^2 + \ldots + a_{k-1} o_m(1)^{k-1}. \tag{2}$$

Then form n equations to form k element equations. If $n > k$, equation has a solution, then

$$\begin{pmatrix} o_m^{k-1}(1) \\ o_m^{k-1}(2) \\ \vdots \\ o_m^{k-1}(n) \end{pmatrix} = \begin{pmatrix} 1 & o_m(1) & \cdots & o_m(1)^{k-1} \\ 1 & o_m(2) & \cdots & o_m(2)^{k-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & o_m(n) & \cdots & o_m(n)^{k-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{k-1} \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_{k-1} \end{pmatrix},$$
$$O_{n\times 1} = o_{n\times k} \cdot A_{k\times 1} + E_{n\times 1}. \tag{3}$$

The matrix is expressed as

A is the least square fitting solution of different windows, and the value O_m' is as follows:

$$O_m' = O_m \cdot A. \tag{4}$$

Smoothing dataset is $O_m' = (o_m'(1), o_m'(2), \ldots, o_m'(n))$

2.2.4. GMM Classification. Gaussian mixture model (GMM) is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters [54],

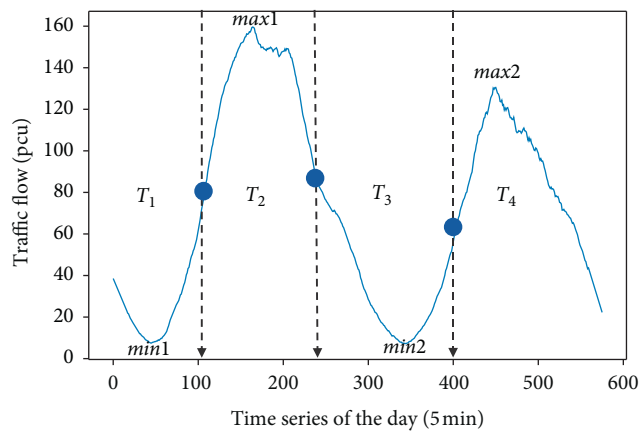


Figure 3: Time series division diagram.

which is used to judge the correlation between upstream and predicted traffic flow, while EM is used to obtain the maximum likelihood estimation of GMM [55]. In this paper, KL-divergence and GRA coefficient are two parameters to do the binary classification.

(1) KL Divergence. KL divergence [56], which is also known as relative entropy, is broadly used as the measurement of the dissimilarity between two probabilistic models [57]. Define KL divergence between $O_i = (o_i(1), o_i(2), \ldots, o_i(n))$ and $D_0 = (d_0(1), d_0(2), \ldots, d_0(n))$ as

$$\mathrm{KL}(O_i D_0) = \sum_{j=1}^{n} o_i(j) \log_2 \frac{o_i(j)}{d_0(j)}. \tag{5}$$

The closer KL divergence is to 0, the more similar the two distributions are.

(2) GRA Coefficient. GRA is a method to judge the similarity of different datasets. Compared with traditional Pearson correlation [58], it can use a smaller amount of data to reflect the linear similarity between traffic flows [59]. The steps of GRA are as follows.

Since there is little difference in the magnitude of traffic flow at the same point, the data are initialized by dividing the initial value of the flow $o_i(1)$ and $d_0(1)$.

$$\begin{aligned} o_i(k)_{\mathrm{GRA}} &= \frac{o_i(k)}{o_i(1)}, \\ d_0(k)_{\mathrm{GRA}} &= \frac{d_0(k)}{d_0(1)}, \end{aligned} \tag{6}$$

where $o_i(k)$ denotes traffic flow of upstream point O_1 in the k time and $d_o(k)$ denotes traffic flow of prediction point D_0 in the k time.

Define the prediction sequence $(d_0(1)_{\mathrm{GRA}}, d_0(2)_{\mathrm{GRA}}, \ldots, d_0(n)_{\mathrm{GRA}})$ as D_{GRA} and the factor sequence $(o_i(1)_{\mathrm{GRA}}, o_i(2)_{\mathrm{GRA}}, \ldots, o_i(n)_{\mathrm{GRA}})$ as. $O_{i\mathrm{GRA}}$

Then calculate the relation coefficient in the k time

$$\gamma(d_0(k)_{\mathrm{GRA}}, o_i(k)_{\mathrm{GRA}}) = \frac{\min_i \min_k |d_0(k)_{\mathrm{GRA}} - o_i(k)_{\mathrm{GRA}}| + \xi \max_i \max_k |d_0(k)_{\mathrm{GRA}} - o_i(k)_{\mathrm{GRA}}|}{|d_0(k)_{\mathrm{GRA}} - o_i(k)_{\mathrm{GRA}}| + \xi \max_i \max_k |d_0(k)_{\mathrm{GRA}} - o_i(k)_{\mathrm{GRA}}|}, \tag{7}$$

where ξ denotes the coefficient to control the degree of differentiation, which is generally 0.5 [58].

Define the mean value as GRA coefficient of D_0 and O_i:

$$\mathrm{GRA}(D_0, O_i) = \frac{1}{n} \sum_{k=1}^{n} \gamma(d_0(k), o_i(k)). \tag{8}$$

(3) GMM Classification. Let $x_{O_i} = (\mathrm{KL}_{O_i}, \mathrm{GRA}_{O_i})$, $x = (x_{O_1}, x_{O_2}, x_{O_3}, \ldots, x_{O_n})$. GMM is classified by calculating probability. Two-dimensional Gaussian mixture model is as follows:

$$p(x) = \sum_{k=1}^{K=2} \frac{\vartheta_k}{2\pi^{n/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_k)^{\mathrm{T}} \sum_{k}^{-1} (x-\mu_k)\right], = \sum_{k=1}^{K=2} \vartheta_k \mathscr{R}(x|\mu_k, \Sigma_k), \tag{9}$$

where μ_k denotes expectation, Σ_k denotes covariance, n denotes data dimension, $\mathscr{R}(x|\mu_k, \Sigma_k)$ denotes k component in a hybrid model, ϑ_k denotes mixture coefficient and

$$\sum_{k=1}^{k=2} \vartheta_k = 1, \quad 0 < \vartheta_k < 1. \tag{10}$$

Then use EM algorithm to calculate $\vartheta_1, \mu_1, \Sigma_1, \vartheta_2, \mu_2, \Sigma_2$. EM-GMM pseudocode is as follows: (Algorithm 1)

2.3. TGWO-LSTM-KNN Hybrid Model. TGWO-LSTM-KNN hybrid model consists of LSTMs module, KNN module, and TGWO module. The specific process is shown in Figure 4.

The TGWO-LSTM-KNN model first splits the dataset. The traffic flow of $O_1 \sim O_m$ is trained in the LSTMs module to predict future traffic flow $P_1 \sim P_m$ in parallel. Then use traffic flow dataset consisting of $O_1 \sim O_m$, $P_1 \sim P_m$, and D_0 to predict D_0^p in the KNN module. The step and epochs of

function EM − GMM $(k \longleftarrow 2,\ \vartheta_1, \mu_1, \Sigma_1, \vartheta_2, \mu_2, \Sigma_2, i \longleftarrow 0)$
  do $i \longleftarrow i + 1$:
  calculate $p(x)$;
  E − step: $\gamma(\beta_{nk}) \longleftarrow \vartheta_k \mathscr{R}(x_{O_n}, |\mu_n, \Sigma_n) / \sum_{j=1}^{K} \vartheta_j \mathscr{R}(x_{O_n}, |\mu_j, \Sigma_j)$;
  M − step: $\mu_k^{\text{new}} \longleftarrow 1/N_k \sum_{n=1}^{N} \gamma(\beta_{nk}) x_{O_n}$;
    $\Sigma_k^{\text{new}} \longleftarrow 1/N_k \sum_{n=1}^{N} \gamma(\beta_{nk})(x_{O_n} - \mu_k^{\text{new}})(x_{O_n} - \mu_k^{\text{new}})^T$;
    $\vartheta_k^{\text{new}} \longleftarrow N_k / N$;
    $N_k \longleftarrow \sum_{n=1}^{N} \gamma(\beta_{nk})$;
  calculate $\ln\ p^i(x|\vartheta, \mu, \Sigma) \longleftarrow \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \vartheta_k \mathscr{R}(x_k, |\vartheta_k, \Sigma_k)\right\}$;
  until $\ln\ p^{i+1}(x|\vartheta, \mu, \Sigma) - \ln\ p^i(x|\vartheta, \mu, \Sigma) < T$;
  return $\vartheta_1, \mu_1, \Sigma_1, \vartheta_2, \mu_2, \Sigma_2$;
end function

Algorithm 1: EM-GMM pseudocode.

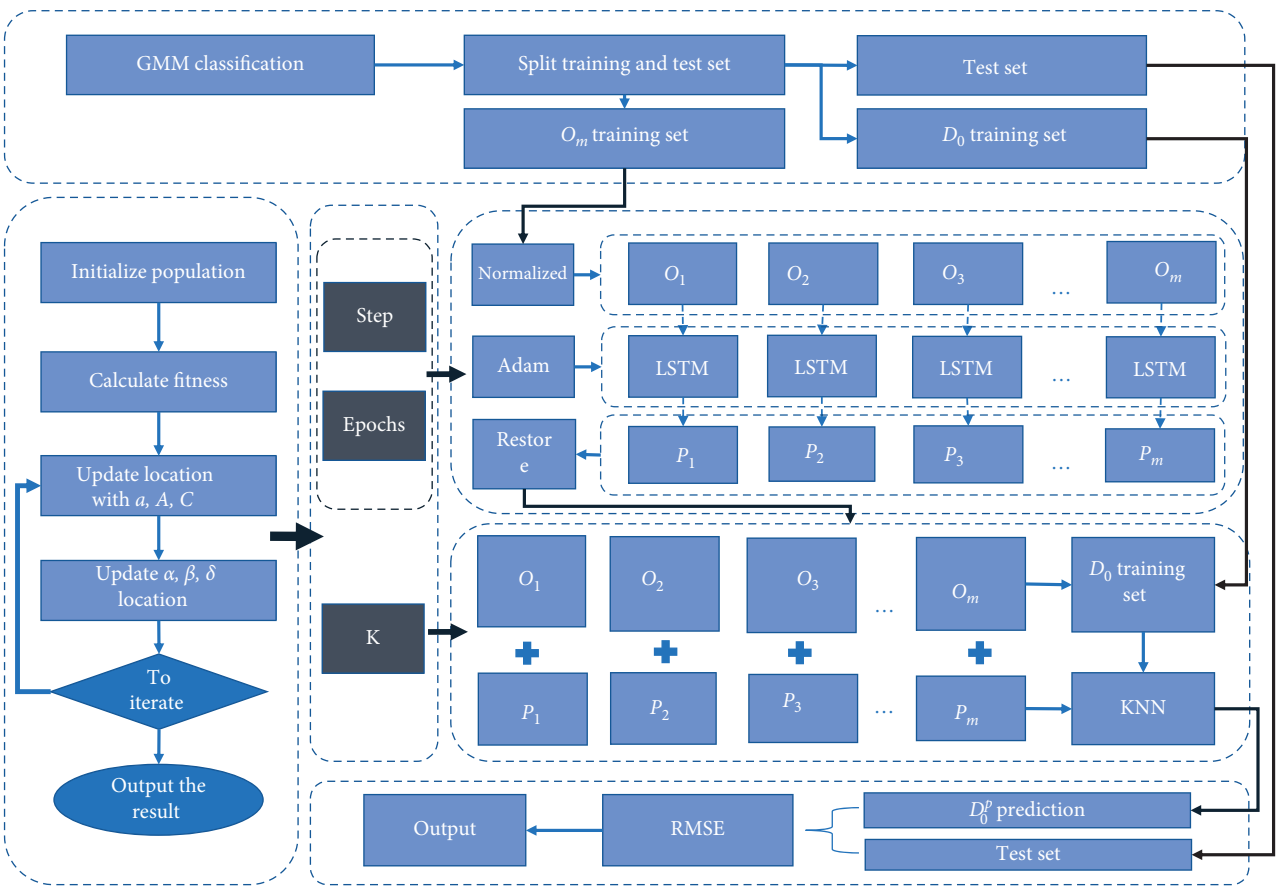


Figure 4: TGWO-LSTM-KNN with GMM classification flowchart.

LSTM are the parameters with great influence [25], and the coefficient K in the KNN module is also one of the key parameters. Then use the TGWO module to optimize these parameters.

2.3.1. LSTMs Module. LSTM is a special RNN [24] with a forgetting gate. The sigmoid function is used to prevent gradient explosion and disappearance. The traffic flow data of upstream points are trained in different LSTM threads in parallel, which is defined as LSTMs. The data input time is also changed to reduce the lag of the module, which makes LSTM more accurate. The memory unit of the module is shown in Figure 5.

The calculation processes of the memory unit $o_m(n)$ and parameters (see Table 3) are as follows:

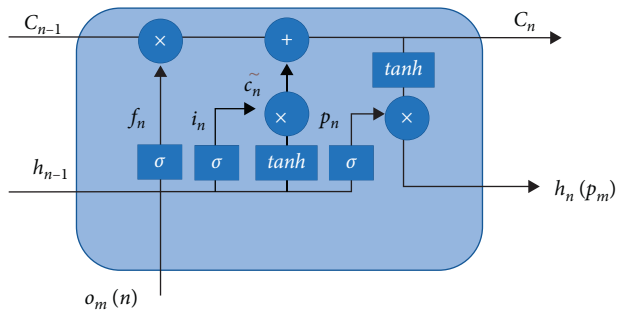


Figure 5: LSTM memory unit [24].

Table 3: LSTM parameters interpretation.

Symbol	Meaning
$\tilde{c}_n$	The updated state of a memory cell
i_n, f_n, p_n, c_n, h_n	Input gate, forgetting gate, output gate, memory cell, hidden layer output value
w_{oc}, w_{ch}	The weight of $o_m(n)$ and the hidden layer
$o_m(n)$	The input of the $n.th$ data at the $m.th$ upstream point
c_{n-1}, h_{n-1}	The output of the memory cell and the hidden layer at $n-1$
w_{oi}, w_{hi}, w_{ci}	The weight of the input gate of $o_m(n)$, the hidden layer, and the memory cell
w_{of}, w_{hf}, w_{cf}	The weight of the forgetting gate of $o_m(n)$, the hidden layer, and the memory cell
w_{op}, w_{hp}, w_{cp}	The weight of the output gate of $o_m(n)$, the hidden layer, and the memory cell
b_c, b_i, b_f, b_p	Offset item
tanh and σ	Activation functions
$// \circ //$	Dot product

$$\begin{aligned}
\tilde{c}_n &= \tanh\left(w_{oc}o_m(n) + w_{ch}h_{n-1} + b_c\right),\\
i_n &= \sigma\left(w_{oi}o_m(n) + w_{hi}h_{n-1} + w_{ci}c_{n-1} + b_i\right),\\
f_n &= \sigma\left(w_{of}o_m(n) + w_{hf}h_{n-1} + w_{cf}c_{n-1} + b_f\right),\\
p_n &= \sigma\left(w_{op}o_m(n) + w_{hp}h_{n-1} + w_{cp}c_n + b_o\right),\\
c_n &= c_{n-1} \circ f_n + i_n \circ \tilde{c}_n,\\
h_n &= p_n \circ \tanh(c_n).
\end{aligned} \tag{11}$$

2.3.2. KNN Module. In this paper, the KNN model is used to predict D_0^p by using data of $O_1 \sim O_m$, $P_1 \sim P_m$, and D_0. This method not only has high efficiency and less complexity but also can meet the needs of multifactor. Euclidean distance is calculated as follows [14]:

$$d_x = \sqrt{\sum_{i=1}^{m} \left(p_i - o_i(x)\right)^2}, \tag{12}$$

where d_n denotes the Euclidean distance between P_i in the current time and the $O(x)$ vector x time, P_i denotes the current traffic flow vector of different upstream points $(p_1, p_2, \ldots, p_m)$ in the prediction dataset, p_i denotes the current value of the ith upstream point in P_i, $O(x)$ denotes x time traffic flow vector of different upstream points $(o_1(x), o_2(x), \ldots, o_m(x))$, $o_i(x)$ denotes the value of ith upstream point in $O(x)$, and m denotes the number of upstream points.

To sort d_x in ascending order, define K smallest $O(x)$ as $O(x_1), O(x_2), \ldots, O(x_k)$ The corresponding values of its prediction point are $d_0(x_1), d_0(x_2), \ldots, d_0(x_k)$. Predict d_0^p with weighted method and define prediction set of d_0^p at different time as D_0^p.

$$\begin{aligned}
w_i(x_n) &= \frac{O(x_n)}{\sum_{n=1}^{k} O(x_n)},\\
d_0^p &= \sum_{n=1}^{k} w_i(x_n) d_0(x_n),
\end{aligned} \tag{13}$$

where $w_i(x_n)$ denotes weight and d_0^p denotes prediction. LSTM-KNN pseudocode is shown in Algorithm 2.

The number of units in the hidden layer is n_{unit}, and the data dimension D of LSTM in each thread is 1, so the time complexity is $4 \times (n_{\text{unit}}^2 + 2n_{\text{unit}})$. On account of the parallel calculation structure, the total time does not change too much with the increasing number of threads. Compared with the original complexity $4 \times (n_{\text{unit}}^2 + n_{\text{unit}} \times D + n_{\text{unit}})$, which reduces a lot of computation and improves efficiency. The time complexity of KNN is $O(n)$, which is only related to the number of data, so the calculation speed is fast.

LSTM has good robustness in traffic prediction [60] and also improves performance by setting reasonable time lag [61] or forgetting layer. KNN can avoid large deviation directly because it calculates the closest Euclidean distance to produce the prediction [62]. These two positive aspects of robustness will support the proposed method to gain more adaptability on data fluctuations and environmental change.

2.3.3. TGWO Module. Grey wolves algorithm was put forward by scholars Mir Jalili Australia in 2014, the Grey wolf groups according to social relations are divided into four grades. Each wolf represents a candidate solution, while the most optimal solution is α_T, the suboptimal solution is β_T, the third optimal solution is δ_T, and the last is ω_T. In each iteration, the three optimum solutions as α_T, β_T, δ_T

```
function LSTM_KNN (step_1, ..., step_n, epochs_1, ..., epochs_n, k):
  transform dataset into MinMaxScaler (feature_range = (0, 1));
  spilt dataset ['O_n'] into O_n train and O_n test;
  LSTMs − step:
  thread_1: look_back ⟵ step_1; reshape O_1train and O_1test into LSTM_dataset with look_back;
    model_1.add (LSTM (n_unit));
    model_1.dense (1);
    model_1.fit (O_1train, epochs_1, batch_size ⟵ number of days);
    O_1prediction ⟵ model_1.predict (O_1test);
  return O_1prediction;
  ⋮
  thread_n: look_back ⟵ step_n; reshape O_n train and O_n test into LSTM_dataset with look_back;
    model_n.add (LSTM (n_unit));
    model_n.dense (1);
    model_n.fit (O_n train, epochs_n, batch_size ⟵ number of days);
    O_1prediction ⟵ model_1.predict (O_1test);
  return O_n prediction;
  KNN − step:
    knn_test ⟵ stack O_1prediction, ..., O_n prediction in sequence horizontally;
    knn_train ⟵ dataset;
    knn.fit (knn_train ['O_n'], knn_train ['D_0']);
    knn_prediction ⟵ knn.predict (knn_test, k);
  restore knn_prediction;
  return knn_prediction;
end function
```

Algorithm 2: LSTM-KNN pseudocode.

determine the prey position and direct the ω_T to update the position around it [63].

By using the improved adaptive convergence factor [64], the extremum can be quickly found when the step size is large in a global search. Besides, the extremum can be prevented from missing when the step size becomes smaller in the local search. The weight step size formula [64] adds the weight decreasing strategy, which can reduce unnecessary iterative processes and improve efficiency. The calculation method is as follows.

(1) Initialize the Population. The upper bound U_b and lower bound L_b are defined, respectively. The number of wolves is N. The dimensions are S. $M_{N\times S}$ denotes an $N \times S$ two-dimensional matrix, which is the searching field. There are $2m+1$ key parameters to be optimized in each element of the field array, which are step, epochs in the LSTMs module, and the K value in the KNN module. Generate integers randomly at their respective upper and lower bounds to form an element of field array:

$$M_{[i,j]} = [\text{step}_1, \text{step}_2, \ldots, \text{step}_n, \text{epochs}_1, \text{epochs}_2, \ldots, \text{epochs}_n, k]. \tag{14}$$

Each parameter has a different U_b and L_b. Set up two vectors to record the bounds.

$$U_{\text{TGWO}} = [U_1, U_2, \ldots, U_{2m+1}], L_{\text{TGWO}} = [L_1, L_2, \ldots, L_{2m+1}]. \tag{15}$$

(2) Calculate Fitness. Input the element corresponding to each wolf into LSTM-KNN and compare the error. Define the optimal solution as α_T, β_T, δ_T, respectively.

(3) Update location with a_T, A_T, C_T:

$$\begin{aligned}
D_\alpha &= \left|C_T \cdot W_\alpha(t) - W(t)\right|,\\
D_\beta &= \left|C_T \cdot W_\beta(t) - W(t)\right|,\\
D_\delta &= \left|C_T \cdot W_\delta(t) - W(t)\right|,\\
W_\alpha(t+1) &= \varphi W_\alpha(t) - A_T \cdot D_\alpha,\\
W_\beta(t+1) &= \varphi W_\beta(t) - A_T \cdot D_\beta,\\
\varphi(t+1) &= \varphi_{\max} - (\varphi_{\max} - \varphi_{\min}) \cdot \frac{t}{t_{\max}},\\
W(t+1) &= \frac{W_\alpha(t+1) + W_\beta(t+1) + W_\delta(t+1)}{3},\\
A_T &= 2a_T(r_1 - 1),\\
C_T &= 2r_2,\\
a_T &= a_{\max} - \left[1 + \exp\left(-\frac{t}{t_{\max}}\right)\right],
\end{aligned} \tag{16}$$

where D denotes the distance between the grey wolves and their prey, t denotes current iteration times, $W_P(t)$ denotes the position of the grey wolf in t iteration times, $W(t)$ denotes the position of the prey in t iteration times, A and C denote the coefficient vectors, r_1 and r_2 denote random coefficients with scalars between 0 and 1, generally take 0.5, a denotes the convergence factor, φ denotes the weight of

inertia, $\varphi_{\max}$ denotes the maximum weight of inertia, generally take 0.9, and $\varphi_{\min}$ denotes the minimum weight of inertia, generally take 0.4.

(4) Complete the Iteration and Output the Result. The optimal solution α_T is denoted as M_{best}. The global optimal solution is as follows:

$$M_{\text{best}} = [\text{step}_1, \text{step}_2, \ldots, \text{step}_n, \text{epochs}_1, \text{epochs}_2, \ldots, \text{epochs}_n, n_1, n_2, \ldots, n_n, k]. \tag{17}$$

3. Experiments and Analysis

3.1. Experimental Data. Whitemud Drive is an in-city highway across Edmonton, Alberta, Canada. It is 28 kilometers long with a basic speed limit of 80 kilometers per hour. As a test road, Whitemud Drive is equipped with seven traffic video cameras and seven loop detectors (VDS1017, VDS1037, VDS1034, VDS1031, VDS1029, VDS1027, and VDS1019) from west to east on the main road and gate road to observe the vehicle flow, the vehicle speed, and the vehicle density. In this paper, data of 15 working days are used as historical data for experiments.

3.2. GMM Selecting Test. VDS1019 is set as the prediction point D_0, and the change of traffic flow within one day of the working day is plotted according to 5 mins (see Figure 6). To better carry out the time-division work, the data are smoothed by SG, and the image is reconstructed (see Figure 7).

Find the extreme values, set up the midpoints, and divide the time series (see Figure 8).

The time-division results are shown in Table 4.

3.2.1. Reconstructing the Dataset by Time-Division. VDS1017, VDS1037, VDS1034, VDS1031, VDS1029, and VDS1027 are recorded as $O_1 \sim O_6$, their historical data on working days are divided into parts according to T1 ~ T4, and the data in the same time part are put into the same column. Since the length of the road is 28 km and the speed limit is 80 km/h, we choose 60 km/h as the test speed. It takes 30 mins to go from VDS1017 to VDS1019. Considering the continuity of the road network, the vehicle passes through each point every 5 mins, so $O_2 \sim O_6$ is delayed by 5–25 mins for input. And the prediction point D_0 is delayed by 30 mins for input. In this way, the data of each day are the delayed input to form a dataset.

Calculate KL divergence and GRA coefficient (see Table 5).

Input the result table into the GMM module for classification.

Taking T2 as an example, the GMM classification results are shown in Figure 9.

The classification results are two types. The closer the GRA coefficient is to 1, the better the correlation is, and the closer the KL divergence is to 0, the more similar the distribution is. So, choose the yellow mark points as the input points (see Figure 9). The following is the final classification results of four datasets (see Table 6).

The upstream points selected between T2 and T3 are the same, so T2 and T3 can be regarded as the same time part, which is 7 : 00–13 : 00, and the dataset T_{2+3} can be reconstructed. The next experimental data are T_{2+3}.

3.3. Comparative Experiments at Different Upstream Points. Different upstream points are selected for model prediction and mean absolute percentage error (MAPE) comparison so as to verify the effect after classification. Results are shown in Table 7.

In this dataset, it can be seen that the abandoned upstream points have little difference from the selected points (see Table 5). So it is not obvious in the accuracy improvement, which is reasonable. If in datasets with a large difference, the meaning of GMM classification operation will be reflected more.

3.4. TGWO-LSTM-KNN Experiments

3.4.1. LSTM-KNN Structural Test. Considering the operation time and efficiency, this paper constructs the following structures for testing. Since the training data are 15 days' traffic flow, the batch size is set to 15, and the data are divided into 15 groups, which will ignore the data relationship between each group and reduce the risk of overfitting. If the batch size is too small, it is not conducive to the training model and is easy to overfitting. The test step is 3, epochs are 100, and the comparison results are shown in Table 8.

The results show that increasing the number of LSTM layers can improve the prediction accuracy, but it increases a lot of computing time and overfits easily. Increasing the forgetting layer (the forgetting rate is 0.2) will reduce the accuracy. In the first-layer structure, there is only a small difference between 256 units and 128 units; therefore, the single-layer 128 units structure is selected for prediction in the following LSTM.

3.4.2. LSTM-KNN Time Steps Test. The model is tested under different time steps (5 mins, 10 mins, 15 mins, and 30 mins). The result shows that the model has good accuracy and stability (see Figure 10). The overall prediction accuracy shows a downward trend, and the absolute error shows an upward trend. It is worth noting that when the time step is 10 mins, the trend of accuracy will change and the error is lower than 15 mins. In general, even if the time step is different, the model still has good performance for the prediction accuracy.

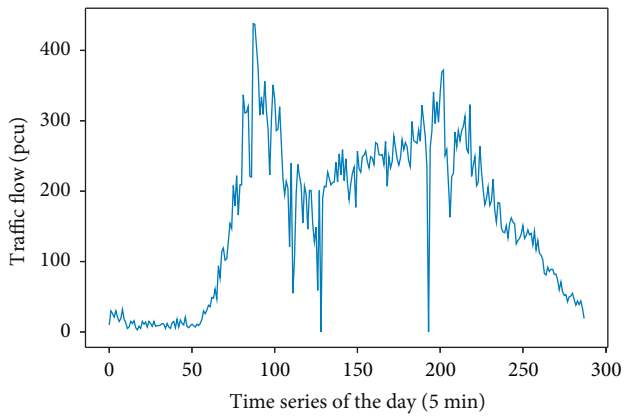


Figure 6: VDS1019 traffic flow before SG filter.

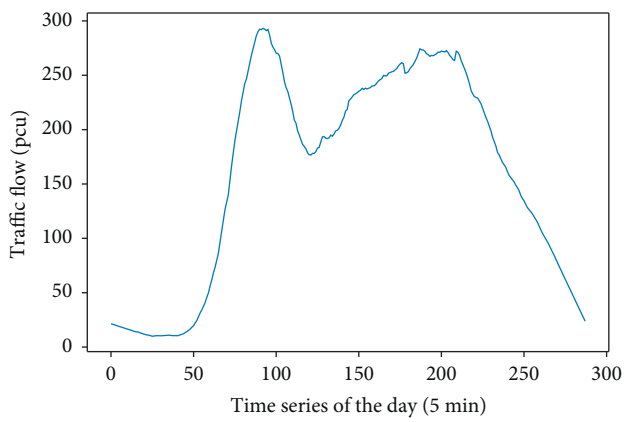


Figure 7: VDS1019 traffic flow after SG filter.

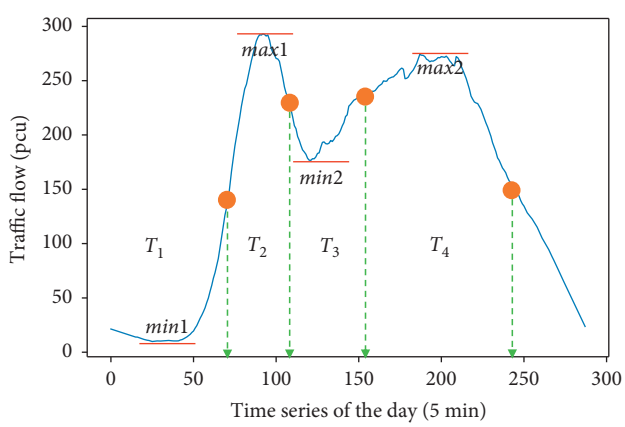


Figure 8: VDS1019 time-division diagram.

3.4.3. TGWO Optimization Test. The dataset has four upstream points, and the key parameters to be optimized are the steps of O_2, O_4, O_5, O_6, the training epochs in the LSTM module, and the K in the KNN module. So, $M_{[i,j]} = [\text{step}_2, \text{step}_4, \text{step}_5, \text{step}_6, \text{epochs}_2, \text{epochs}_4, \text{epochs}_5, \text{epochs}_6, k]$. Set the upper and lower bounds: the step size ranges from 3 to 20. Training time ranges from 100 to 200. The coefficient K ranges from 5 to 50, the number of iterations t is 50, the number of wolves is $N = 5$, and the dimension S is 30. The training results are shown in Figure 11.

When the MAPE is 10.04 (green mark in Figure 11), it will reach the optimal solution $M_{\text{best}} = [5, 6, 5, 6, 120, 100, 120, 100, 25]$. The steps of O_2, O_5 and O_4, O_6 are 5 and 6, the training *epochs* are 120 and 100, and K is 25.

3.4.4. Comparison of TGWO-LSTM-KNN and Other Models. TGWO-LSTM-KNN is compared with SVM, LSTM, and BP. The parameters of TGWO-LSTM-KNN are the optimal

Table 4: Time-division results.

Time part name	Number of data	Time
T_1	0–70	0 : 00–7 : 00
T_2	71–110	7 : 00–9 : 00
T_3	111–155	9 : 00–13 : 00
T_4	156–249	13 : 00–20 : 00

Table 5: Calculation results of KL divergence and GRA coefficient.

KL/GRA result	T1		T2		T3		T4	
	KL	GRA	KL	GRA	KL	GRA	KL	GRA
O_1	0.030	0.90	0.023	0.78	0.020	0.86	0.010	0.87
O_2	0.029	0.90	0.009	0.81	0.017	0.88	0.012	0.86
O_3	0.038	0.89	0.021	0.78	0.031	0.85	0.035	0.82
O_4	0.035	0.90	0.009	0.81	0.018	0.88	0.011	0.85
O_5	0.027	0.91	0.008	0.83	0.016	0.89	0.009	0.87
O_6	0.008	0.95	0.005	0.88	0.004	0.93	0.004	0.92
D	0	1	0	1	0	1	0	1

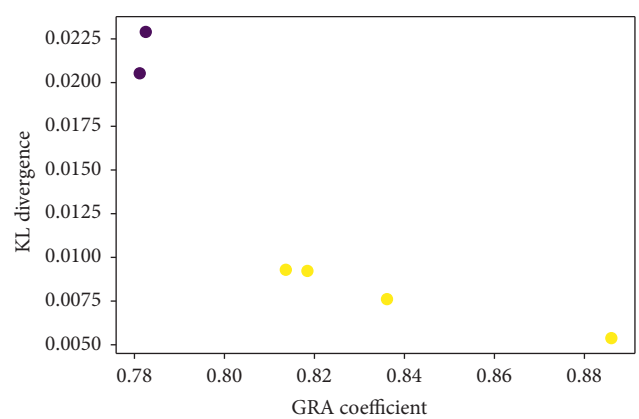


Figure 9: GMM for T2 dataset classification result.

Table 6: Classification results of GMM.

GMM result	T1	T2	T3	T4
D point	O_6	O_2, O_4, O_5, O_6	O_2, O_4, O_5, O_6	O_1, O_2, O_4, O_5, O_6

Table 7: MAPE comparison table of different points.

Model/points	V34	V34 + V31	V34 + V31 + V29	V34 + V31 + V29 + V27	Including abandonment points
LSTM-KNN	14.45	14.96	14.31	12.19	12.46
Linear-SVM	20.47	21.14	20.03	21.17	21.24
Poly-SVM	15.85	17.30	17.15	17.65	18.04
RBF-SVM	20.51	20.29	20.87	20.17	20.48
Multi-LSTM	14.32	14.51	11.96	9.51	23.14

Table 8: Error comparison table of different structures.

Structure	RMSE	MAE	MAPE
Double-layer structure LSTM (256 units) + LSTM (128 units)	42.49	29.81	11.64
Four-layer structure LSTM (256 units) + Dropout (0.2) + LSTM (128 units) + Dropout (0.2)	44.87	31.33	12.25
Single-layer structure LSTM (256 units)	44.42	31.55	12.32
Single-layer structure LSTM (128 units)	44.43	31.29	12.19

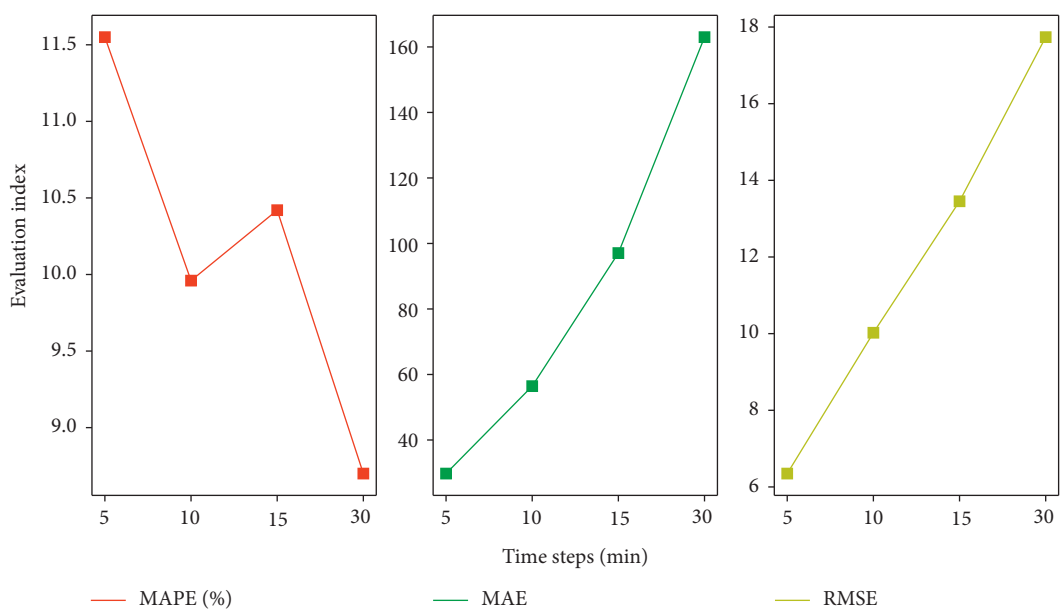


Figure 10: Different error (MAPE, MAE, RMSE) comparisons of different time steps.

solution. The step of LSTM-KNN is 3, K is 16, and the epochs is 100. LSTM has 256 units in the first layer and 128 units in the second layer of the double-hidden layer structure, with the step of 3 and epochs of 100. SVM uses three modes to fit, linear and poly, RBF. BP neural network is a three-layer structure, two fully connected layers, the middle increased the forgetting layer (rate is 0.2), and the epochs is 100. Results are shown in Figure 12.

The accuracy of LSTM-KNN can reach the level of the popular model. The accuracy of TGWO-LSTM-KNN can be improved by 15.27% compared with single-LSTM, 9.47% compared with BP, and 43.12% compared with poly-SVM (see Table 9). Besides, the advantage of the hybrid model is not accuracy, but being able to output the prediction set of upstream points for collaborative management.

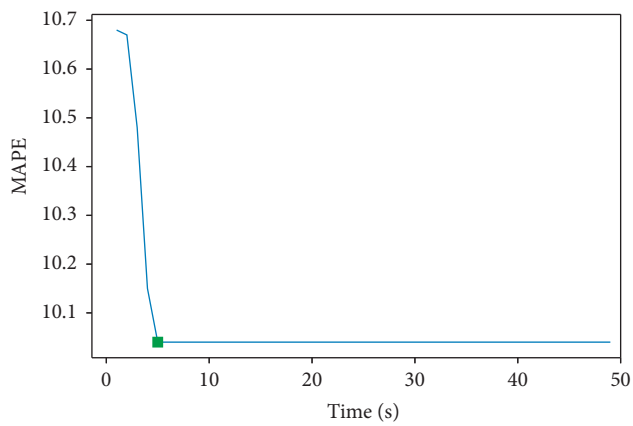


Figure 11: TGWO iterative convergence diagram.

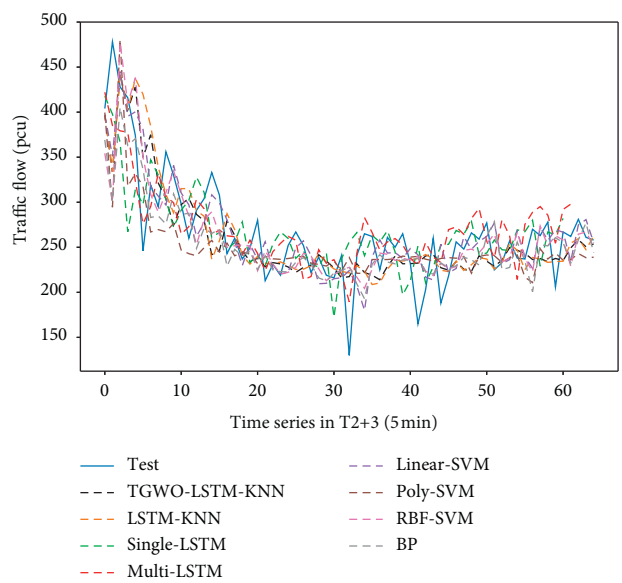


Figure 12: Model comparison diagram.

Table 9: Error comparison table of different models.

Model	RMSE	MAE	MAPE
TGWO-LSTM-KNN	29.70	23.17	10.04
LSTM-KNN	44.43	31.29	12.19
Single-LSTM	36.75	27.88	11.85
Multi-LSTM	33.32	24.52	9.51
BP	37.59	27.59	11.09
Linear-SVM	38.78	27.03	21.17
Poly-SVM	45.36	33.87	17.65
RBF-SVM	40.62	28.29	20.17

4. Conclusions

In this paper, the TGWO-LSTM-KNN prediction model with GMM classification considering spatial-temporal characteristics under the concept of attention mechanism is proposed. The time series is divided into parts by using the temporal characteristic of the prediction point. And GMM through KL and GRA is used for further classification. The upstream points with a small difference in distribution and high linear similarity are selected to increase the accuracy and reduce the complexity. Then the hybrid model TGWO-LSTM-KNN is used to train and predict. Parallel computing is used in the LSTM module to improve efficiency.

GMM as an unsupervised model can be very flexible to classify. This model can be applied to the multifactor model to reduce complexity. KNN, as the next part of LSTM, fully combines upstream points data and prediction points data for prediction. Compared with SVM, KNN has the characteristics of fast speed and greater data processing ability, which is more suitable for multiple factors and complex data of freeway. LSTMs module can ignore the relationship between upstream points to perform the parallel computation. It can reduce the operation time and make the model more practical. TGWO is less likely to fall into local optimal solution and also has fast speed and good performance. To sum up, TGWO-LSTM-KNN with GMM classification can be better used in real freeways with complex data and multifactor with high accuracy, fast calculation speed, and strong adaptability. It can be applied in the real freeway to achieve the purpose of collaborative management.

Data Availability

The data used to support the findings of this study are openly available in the OpenITS platform for noncommercial purposes only and the website is https://www.openits.cn/openData1/700.jhtml.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors want to thank Tianjiao Wang, Xi Chen, Xiang Wang, and Lingbin Kong for their help in the study. Thanks also should be given to the OpenITS platform (http://www.openits.cn/). The research and publication of this article were funded by the National Natural Science Foundation of China (52072129).

References

[1] C. Qi and Z.-S. Hou, "Application of adaptive single-exponent smoothing for short-term traffic flow prediction," *Control Theory & Applications*, vol. 29, no. 4, pp. 465–469, 2012.

[2] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.

[3] H. Gao and D. Zhang, "Traffic flow forecasting model based on fractal and three-exponential smoothing," *Journal of Nanjing University of Posts and Telecommunications (Natural Science Edition)*, vol. 38, no. 6, pp. 63–67, 2018.

[4] L. Lv, M. Chen, Y. Liu, and X. Yu, "A plane moving average algorithm for short-term traffic flow prediction," in *Advances in Knowledge Discovery and Data Mining*, T. Cao, E. P. Lim, Z. H. Zhou, T. B. Ho, D. Cheung, and H. Motoda, Eds., pp. 357–369, Springer, Cham, Switzerland, 2015.

[5] T. Mai, B. Ghosh, and S. Wilson, "Short-term traffic-flow forecasting with auto-regressive moving average models," *Proceedings of the Institution of Civil Engineers - Transport*, vol. 167, no. 4, pp. 232–239, 2014.

[6] D. Billings and Y. Jiann-Shiou, "Application of the ARIMA models to urban roadway travel time prediction - a case study," in *Proceedings of the 2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1-6, p. 2529, Toronto, ON, Canada, October 2006.

[7] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, 2015.

[8] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *Journal of Highway and Transportation Research and Development*, vol. 3, pp. 82–85, 2004.

[9] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[10] L. L. Ojeda, A. Y. Kibangou, and C. C. De Wit, "Adaptive kalman filtering for multi-step ahead traffic flow prediction," in *Proceedings of the 2013 American Control Conference*, pp. 4724–4729, Washington, DC, USA, June 2013.

[11] Z. Liyan, M. Jian, and S. Jian, "Examples of validating an adaptive kalman filter model for short-term traffic flow prediction," in *Proceedings of the Twelfth COTA International Conference of Transportation Professionals*, Xi'an, China, December 2012.

[12] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, and J. Qin, "Hybrid dual Kalman filtering model for short-term traffic flow forecasting," *IET Intelligent Transport Systems*, vol. 13, no. 6, pp. 1023–1032, 2019.

[13] S. Zhang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "Noise-identified kalman filter for short-term traffic flow forecasting," in *Proceedings of the 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, Shenzhen, China, Decmber 2019.

[14] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[15] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *Journal of Transportation Engineering*, vol. 132, no. 2, pp. 114–121, 2006.

[16] Y. Zhang and Y. Xie, "Forecasting of short-term freeway volume with v-support vector machines," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2024, no. 1, pp. 92–99, 2007.

[17] Y. Sun, B. Leng, and W. Guan, "A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system," *Neurocomputing*, vol. 166, pp. 109–121, 2015.

[18] D. Min, "Short-time prediction of traffic flow based on PSO optimized SVM," in *Proceedings of the 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, Xiamen, China, January 2018.

[19] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction models-a comparison of neural network and nonparametric regression approaches," in *Proceedings of the Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Bari, Italy, October 1994.

[20] S. Dehuri and S.-B. Cho, "A comprehensive survey on functional link neural networks and an adaptive PSO-BP learning for CFLNN," *Neural Computing & Applications*, vol. 19, no. 2, pp. 187–205, 2010.

[21] S. Wang, N. Zhang, L. Wu, and Y. Wang, "Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method," *Renewable Energy*, vol. 94, pp. 629–636, 2016.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[23] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, no. 1, pp. 115–143, 2003.

[26] M. Sundermeyer, R. Schlueter, and H. Ney, *LSTM Neural Networks for Language Modeling*, Interspeech 2012, Portland, OR, USA, 2012.

[27] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM neural network for text classification," *Computer Science*, vol. 1, no. 4, pp. 39–44, 2015.

[28] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[29] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Wuhan, China, November 2016.

[30] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *Ieee Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[31] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *Proceedings of the 2016 Ieee Conference on Computer Vision and Pattern Recognition*, pp. 961–971, Las Vegas, NV, USA, June 2016.

[32] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale Acoustic modeling," in *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, pp. 338–342, Graz, Austria, September 2014.

[33] H. Wen, D. Zhang, and L. Siuyan, "Application of GA-LSTM model in highway traffic flow prediction," *Journal of Harbin Institute of Technology*, vol. 51, no. 9, pp. 81–87+95, 2019.

[34] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of Clinical Epidemiology*, vol. 49, no. 11, pp. 1225–1231, 1996.

[35] J. Guo, Y. Liu, and L. Ma, *Assignee. EEMD-IAGA-BP Neural Network Based Ship Traffic Flow Predicting Method, Involves Constructing Three-Layer BP Neural Network with Enhanced Adaptive Genetic Algorithm Optimization as Training Model, and Obtaining Predicted Results Patent CN110111606-A*, Inventors; Univ Shanghai Maritime, Shanghai Shi, China, 2021.

[36] S. Y. Liong and C. Sivapragasam, "Flood stage forecasting with support vector machines," *Journal of the American Water Resources Association*, vol. 38, no. 1, pp. 173–186, 2002.

[37] M. Duo, Y. Qi, G. Lina, and E. Xu, "A short-term traffic flow prediction model based on EMD and GPSO-SVM," in *Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, March 2017.

[38] H. Chung and K.-s andShin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability*, vol. 10, no. 10, 2018.

[39] W. Zhou and S. Meng, *Assignee. Navigation Reminder Method Based on SVD-PSO-LSTM for Predicting Short-Term Traffic Flow, Involves Providing First Train and Optimizes LSTM Model, Then Collects Historical Traffic Flow Data, and Preprocesses and Input to Training Flow Data Patent CN111709549-A*, Inventors; Univ Donghua, Shanghai, China, 2021.

[40] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *Journal of Advanced Transportation*, vol. 2019, 10 pages, 2019.

[41] L. Dai, H. Mei, C. Qian, M. Yun, and L. Jin-Ming, "Survey on short-term traffic flow forecasting based on deep learning," *Computer Science*, vol. 46, no. 3, pp. 39–47, 2019.

[42] F. Su, H. Dong, L. Jia, Z. Tian, and X. Sun, "Space-time correlation analysis of traffic flow on road network," *International Journal of Modern Physics B*, vol. 31, no. 5, 2017.

[43] Q. Liu, Y. Cai, H. Jiang, X. Chen, and J. Lu, "Traffic state spatial-temporal characteristic analysis and short-term forecasting based on manifold similarity," *Ieee Access*, vol. 6, pp. 9690–9702, 2018.

[44] C. Wang, "Attention-based traffic flow prediction and research," East China Jiaotong University, Nanchang, China, Degree Diss, 2020.

[45] L. LI, "Research on spatial temporal prediction model of traffic flow based on attentional mechanism," South China University of Technology, Guangzhou, China, Degree Diss, 2020.

[46] M. Chen, "Research on traffic flow prediction based on improved graph attention network," Shanghai Normal University, Shanghai, China, Degree Diss, 2021.

[47] D. Ma, X. B. Song, J. Zhu, and W. Ma, "Input data selection for daily traffic flow forecasting through contextual mining and intra-day pattern recognition," *Expert Systems with Applications*, vol. 176, 2021.

[48] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2627–2636, 2021.

[49] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Systems with Applications*, vol. 121, pp. 304–312, 2019.

[50] X. Song, W. Li, D. Ma, D. Wang, L. Qu, and Y. Wang, "A match-then-predict method for daily traffic flow forecasting based on group method of data handling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 11, pp. 982–998, 2018.

[51] D. Ma, B. Sheng, X. Ma, and S. Jin, "Fuzzy hybrid framework with dynamic weights for short-term traffic flow prediction by mining spatio-temporal correlations," *IET Intelligent Transport Systems*, vol. 14, no. 2, pp. 73–81, 2020.

[52] S. Hao, D.-H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 287–300, 2019.

[53] J. Steinier, Y. Termonia, and J. Deltour, "Smoothing and differentiation of data by simplified least square procedure," *Analytical Chemistry*, vol. 44, no. 11, pp. 1906–1909, 1972.

[54] D. Reynolds, *Gaussian Mixture Models*, Springer US, Berlin, Germany, 2008.
[55] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.
[56] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
[57] S. Liu, Z. Wen, J. Tao et al., "A data driven method for target and concatenation cost calculation with KL-divergence in Mandarin hybrid speech synthesis," in *Proceedings of the 2014 12th International Conference on Signal Processing*, pp. 572–576, HangZhou, China, October 2014.
[58] N. Gerhard, *Pearson Correlation Coefficient*, p. 132, Springer, Vienna, Austria, 2009.
[59] J. Deng, "Introduction to grey system theory," *Journal of Grey System*, vol. 1, no. 1, pp. 1–24, 1989.
[60] X. Wang, L. Xu, and K. Chen, "Data-driven short-term forecasting for urban road network traffic based on data processing and LSTM-RNN," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3043–3060, 2019.
[61] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos Solitons & Fractals*, vol. 139, 2020.
[62] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61–78, 2016.
[63] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
[64] W. S. Zhang, Z. Q Hao, J. J. Zhu, T. T. Du, and H. M. Hao, "BP neural network model for short-time traffic flow forecasting based on transformed grey wolf optimizer algorithm," *Journal of Transportation Systems Engineering & Information Technology*, vol. 20, no. 02, pp. 196–203, 2020.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5549111, 14 pages
https://doi.org/10.1155/2021/5549111

Review Article

Applications of Deep Learning Techniques for Pedestrian Detection in Smart Environments: A Comprehensive Study

Fen He,[1] **Paria Karami Olia**,[2] **Rozita Jamili Oskouei**,[3] **Morteza Hosseini**,[4] **Zhihao Peng**,[5] **and Touraj BaniRostam**[2]

[1]*School of Information Engineering, Guangzhou Nanyang Polytechnic College, Guangzhou, Guangdong, China*
[2]*Computer Engineering Department, Islamic Azad University, Central Tehran Branch, Tehran, Iran*
[3]*Department of Computer Science and Information Technology, Mahdishahr Branch, Islamic Azad University, Mahdishahr, Iran*
[4]*Department of Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran*
[5]*School of Software, Dalian Neusoft University of Information, Dalian, China*

Correspondence should be addressed to Rozita Jamili Oskouei; rozita2010r@gmail.com

Received 8 February 2021; Revised 1 April 2021; Accepted 24 August 2021; Published 4 October 2021

Academic Editor: Chunjia Han

Intelligent transportation systems have been very well received by car companies, people, and governments around the world. The main challenge in the world of smart and self-driving cars is to identify obstacles, especially pedestrians, and take action to prevent collisions with them. Many studies in this field have been done by various researchers, but there are still many errors in the accurate detection of pedestrians in self-made cars made by different car companies, so in the research in this study, we focused on the use of deep learning techniques to identify pedestrians for the development of intelligent transportation systems and self-driving cars and pedestrian identification in smart cities, and then some of the most common deep learning techniques used by various researchers were reviewed. Finally, in this research, the challenges in each field are discovered, which can be very useful for students who are looking for an idea to do their dissertations and research in the field of smart transportation and smart cities.

1. Introduction

In recent years, intelligent transportation systems have been developed to help reduce the volume of traffic in metropolitan areas, reduce the rate of accidents and injuries and deaths caused by them, reduce fuel consumption, reduce environmental pollution, and so on. These systems use different technologies (including IoT, machine learning and data mining, neural networks, deep learning, and image processing) for various applications. On the other hand, large automotive and technology companies (such as Google and Tesla) are trying to produce self-driving smart cars that can provide safe travel for people when the driver is drowsy and that can save his life as well as passengers' lives, with automatic control car to prevent any accident. These vehicles must be equipped with sensors to sense the environment and identify objects close to the car and inform the driver and also have actuators to perform real-time operations when the driver is drowsy or the driver does not pay attention to hazards to prevent accidents.

Some of the most important problems in creating and developing self-driving cars are as follows:

(i) Lack of accurate patterns to identify pedestrians and roadblocks with very high accuracy in different roads with different light intensities and different image quality

(ii) Error in identifying pedestrians and obstacles in the paths of self-driving cars

(iii) People's distrust of self-driving cars

(iv) Lack of acceptance of people for 24-hour control and monitoring of various urban and interurban roads' paths

(v) Negative effects of dark weather and snow/ice and fog and rain on the quality and performance of

cameras installed on cars, ultimately reducing the accuracy of pedestrian detection and obstacles in these vehicles

(vi) Lack of necessary infrastructure to implement intelligent transportation on all urban and interurban roads

(vii) The need for high investment to implement the necessary infrastructure to implement intelligent transportation and communication systems to connect vehicles to each other (V2V) and vehicles with roadside infrastructure (V2I)

Machine learning (ML) is a field of artificial intelligence that uses statistical techniques to learn hidden patterns from existing data and to make decisions about unseen records. The main task of a machine learner is building a general model on the possible distribution of training examples and then generalizing experience to unseen examples [1]. The learning process depends on the quality of the data displayed. An example is presented in a dataset with different properties. Unfortunately, extracting efficient features can be difficult for some tasks.

Deep learning is an advanced branch of the ML discipline that aims to discover complex representations of simpler representations. Deep learning methods are usually based on artificial neural networks consisting of several hidden layers with nonlinear processing units. The word deep refers to several hidden layers used to change the display of data. Using the concept of feature learning, each hidden layer of neural networks plots its input data in a new display. The layer manages to absorb a higher level of abstraction than the abstract concept in the previous layer. In deep learning architectures, the hierarchy of features learned at multiple levels is finally mapped to the output of the ML work in a single framework. Similar to ML methods, the deep learning architecture is divided into two broad categories: (a) unsupervised learning methods and (b) supervised learning approaches, including deep neural networks.

This research is organized in five sections. In the second part, we provide a complete description of intelligent transportation systems. In the third section, we provide a brief overview of deep learning and some of its applications. In the fourth section, we describe the use of deep learning to identify pedestrians in smart cities and intelligent transportation systems, review some of the research conducted by various researchers in this field, and state the challenges in each area of research. Finally, in the fifth section, we will provide the conclusion.

2. Intelligent Transport Systems (ITS)

Intelligent transportation systems, for automatic road management and real-time operations in the event of an natural accident (such as a mountain fall, avalanche, and icy road floor) or unnatural disasters (such as accidents, road repairs, and car traffic), have been developed. These systems use sensors as a tool to identify and understand the state of the travel environment. The data collected by these sensors using communication technologies (such as WiFi and DSRC) to other vehicles on the route are sent to control centers.

In recent years, organizations responsible for transport management in different countries of the world have shown great attention to the development and use of intelligent transport systems using the creation of intercity networks. The main reason for this is the diverse applications of automotive communication technology in the four areas of safety promotion, mobility improvement, environmental protection, and asset management. Automotive intelligent communication systems provide technical and economic solutions to the transportation challenges of the 21st century. These systems must be able to enable a growing segment of the human population to move freely, without the risk of accidents and with minimal fuel consumption and environmental pollution [2].

Intelligent transportation technology ranges from basic management systems (such as car navigation, traffic signal control systems, variable driving signs, automatic license plate number recognition, or speed camera) to surveillance applications such as advanced CCTV security systems that provide information. From sources such as car park guide information systems, they collect weather information, etc. [2].

Intelligent Transportation System (ITS) means the use of a set of tools, facilities, and expertise such as traffic engineering concepts, software, hardware, and telecommunications technologies in a coordinated and integrated manner to improve the efficiency and security of transportation systems.

Intelligent transportation systems can also be generalized to different modes of transportation, in which, using automated tools and related scheduling, various types of information receiving and processing operations, as well as traffic management and control, are performed. In this system, by limiting the role of human factors in information processing or control and management processes, we improve the quality of decision-making and management processes.

In intelligent transportation systems (ITS), the definition of transportation infrastructure in addition to information and communication technologies leads to the achievement of goals such as improving passenger safety, reducing transportation time, and reducing fuel consumption and wear or tear of car tires. Applications of ITS include accident management, electronic toll collection management, public transportation management, passenger communication management, and traffic flow management [3].

Compared to traditional traffic engineering, the intelligent transportation system (ITS) has created a new transportation system. Due to different national circumstances, the development priorities of these systems are different, and therefore, the content of ITS research is not the same in all countries. In general, ITS uses information, communication, control, computer technology, and other current technologies to create a real-time, accurate, and efficient transportation management system.

2.1. Architecture of Intelligent Transportation Systems. The US Department of Transportation, through RITA Research and Innovation Technology Management, defined a

national architecture for ITS and provided a common structure for designing intelligent transportation systems. The ITS function model (Logic Architecture) provides a functional view of ITS user services. Physical architecture divides the functions defined by logical architecture into classes and subsystems. Figure 1 shows the high-level diagram of the proposed physical architecture (Architecture Development Team 2007a), in which 22 subsystems (white rectangles) are distributed among four classes: passengers, centers, vehicles, and the field or area of operation.

In Figure 1, the communication requirements between these subsystems are supported by four types of communication, which are shown in the form of an oval in the figure: wireless communication over a wide area, fixed point-to-fixed point communication, vehicle-to-vehicle communication, and dedicated short-range communications [4].

The following is a brief description of each of the classes in Figure 1:

(i) Travelers: different services are provided to passengers (including drivers and occupants of cars), which are generally grouped into two categories, which are as follows [4]:

  (i) Support to their remote travel: using the installation of surveillance cameras by the Road and Transportation Administration on various routes inside and outside the city, road and transport managers can monitor the movements of passengers on different routes from the headquarters and in case of any problem, whether it is an accident or a fall of a mountain, and etc., and send a team to the place immediately

  (ii) Access to personal information: it includes monitoring of the crossing of intersections when the traffic lights are red and registering the license plate number of the offending vehicle and then issuing a fine for him

(ii) Control centers including the centers providing necessary and useful information for drivers, traffic management centers, relief and emergency centers, transport management centers, toll collection center, collected data management centers, transport fleet control centers, and road maintenance management

(iii) Vehicles including personal vehicles, emergency vehicles, commercial vehicles, freight vehicles, support vehicles and relief vehicles, or vehicles belonging to the police patrol

(iv) Roadside equipment including equipment installed on the road, equipment related to toll collection, parking management, checking commercial vehicles, and checking the weight of trucks with their load, which, if it is more than a certain weight, should have stopped and in addition to fining them, their burden should also be reduced

2.2. Some Services Provided in Intelligent Transportation Systems. Some of the most common services provided in intelligent transportation systems are briefly described in this section [5].

2.2.1. Accident Management. We divide the stages of accident management into five stages (depending on the type and severity of the accident) in which one or more stages may occur simultaneously [5]:

(i) Detection and notification: it is the detection and notification of accidents that are often used in mobile phones at this stage.

(ii) Verification: the existence of the accident and the exact type and location of the accident (via traffic surveillance cameras) are confirmed.

(iii) Incident site management: it is a complex process that requires careful coordination, communication, and cooperation between the people present on the scene, all supporting institutions and the general public. Important points for proper incident scene management include the following:

  (i) Providing accurate information to the dispatch unit in a timely manner, including the exact location of the accident, the severity of the accident, and so on

  (ii) Establishing evidence in a safe area to minimize oncoming traffic risk given the location of damaged vehicles and rescue personnel

  (iii) Establishing a command system, especially when major events occur

  (iv) Asking for help from cleaning companies if there is a possibility of hazardous substances at the scene

  (v) Public information in case of an accident through mass media, Internet, or SMS to passengers

(iv) Protection of evidence: it includes the protection of evidence at the scene of the accident and potential evidence that may later be used to prosecute the perpetrator or analyze the data in the future. Any suspicious items at the scene (such as guns, bullets, drugs, and alcohol) should also be protected to be handed over to the police on the scene or the highway warden [5].

(v) Hazardous Materials: when hazardous materials are spilled at the scene of an accident, they must first be thoroughly inspected by police dispatchers, and then, the necessary measures must be taken to collect those materials and clean the environment.

(vi) Breakdown and Demobilization: public mobilization to clear the scene and analyze the accident that occurs when all injured people, damaged vehicles, equipment, and debris are removed from the scene. Public mobilization is to create security, expedient, and regular departure of all those present at the

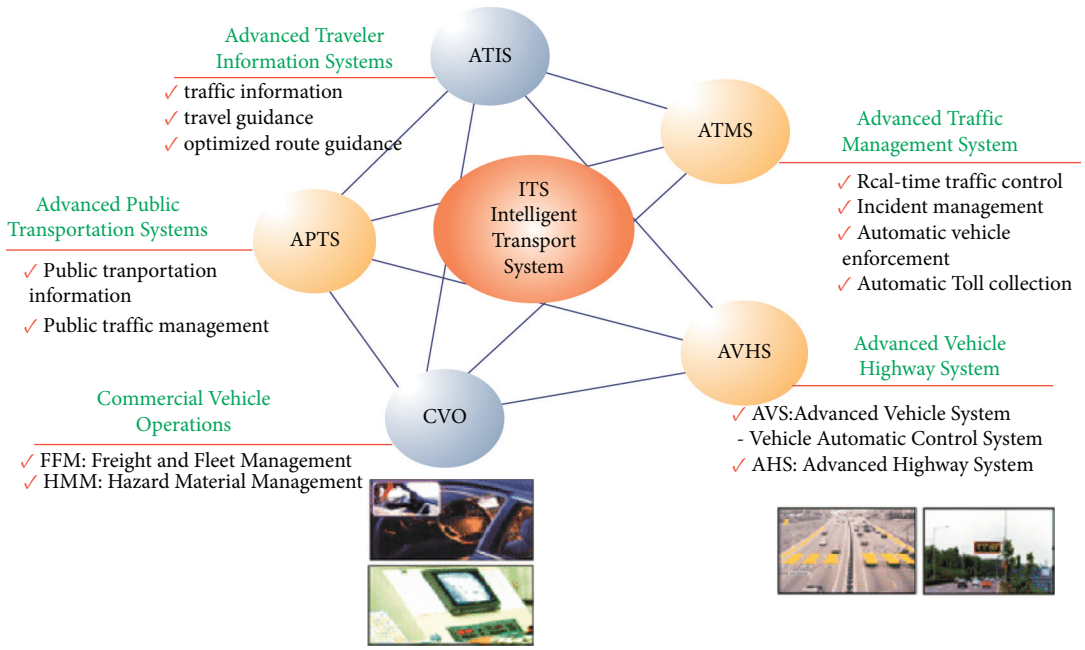


Figure 1: High-level architecture diagram for ITS [4].

scene of the accident and equipment and vehicles from the scene and return the affected area to normal with normal traffic flow [5].

2.2.2. APTS Public Transportation Management. These systems use new information management technologies to increase the efficiency and enhance the security of public transportation systems. These systems include instantaneous and real-time passenger information management systems, vehicle location detection systems, bus arrival time notification systems, and bus crossing priority prioritization systems.

2.2.3. Advanced ATIS Passenger Information Systems. These systems provide information on travel routes and weather conditions for transportation system users, so that they can make the right decisions to choose the route, estimate travel time, and avoid getting caught in crowded routes. Several technologies are used for this purpose, which are as follows [5]:

(i) GPS enabled in car navigation systems

(ii) Dynamic signs and messages for timely and real-time notification in traffic, turns and passes, and accidents or when the road is closed for various reasons such as repairs

(iii) Websites are used to indicate congestion on highways, main streets, and urban and interurban road networks

2.2.4. Advanced ATMS Traffic Management Systems. The data and information obtained through different subsystems (such as vehicle type identifiers, in-vehicle messaging systems, and vehicle connectors) are combined to form a cohesive interface that is capable of parsing and has real-time data analysis and decision-making about the current traffic situation and the assessment of subsequent conditions that may occur, as well as the adoption of appropriate measures to deal with the conditions that have arisen: dynamic traffic control systems, highway operations management systems, accident prevention systems, and making necessary and appropriate decisions when accidents occur, etc. They are considered as advanced traffic management systems [5].

2.2.5. Network Security Management. The main purpose of network safety management, somewhat like the management of sensitive points, means identifying the areas where accidents are most likely to occur. Therefore, there is an urgent need to ensure road safety in those areas. However, there are two important differences between hotspot management and network security management [5]:

(i) In network safety management, the important goal is to identify roads with different degrees of security and ultimately to identify accident hotspots or sensitive points in the road system (such as intersections).

(ii) In network safety management, a report on the severity of accidents is prepared, and accident-prone parts of the road are identified. In the management of critical or accident-prone points, the number of accidents at each critical point is usually too high, so this point is given more importance than the severity of the accident.

3. Deep Learning

Learning is the process by which a system improves its performance by using past experiences. Since 2006, deep learning has emerged as a new subfield of machine learning,

affecting a wide range of signal and information processing in both traditional and modern fields. Many traditional machine learning and signal processing techniques use special architectures that contain a single layer of nonlinear features.

Some examples of deep learning in the workplace include a self-propelled vehicle slowing down as it approaches a pedestrian crossing, an ATM rejecting a counterfeit banknote, and a smartphone app instantly translating an installed signboard performing on the street. Deep learning is especially suitable for identification programs such as face recognition, text translation, voice recognition, and advanced driver assistance systems, including and symptom recognition [6].

3.1. The Difference between Deep Learning and Machine Learning. Deep learning is one of the subfields of machine learning. By learning the machine, the features of an image can be extracted manually. With deep learning, raw images can be inserted directly into a deep neural network that learns features automatically. Deep learning usually requires hundreds of thousands or millions of images to get the best results, while machine learning works well with small datasets. Deep learning is also computationally intensive and requires a high-performance CPU [7].

Deep learning is the most effective, supervised, and cost-effective machine learning approach. Deep learning is not a limited learning method, but it follows a variety of methods and topographies that can be used to make broad predictions about complex problems. This technique includes descriptive and distinctive features in a completely categorized way. Deep learning methods with remarkable performance have achieved significant success in a wide range of applications with useful security tools. Deep learning is used in many applications, including business, comparative experiments, biological image classification, computer insight, cancer detection, natural language processing, object recognition, face recognition, handwriting, speech recognition, stock market analysis, and creation and the development of smart cities.

Machine learning is a subset of artificial intelligence (AI) that gives systems the benefits of automatically learning concepts and knowledge without explicit planning. It begins with observations such as direct experiences to prepare features and patterns in the data and to produce better results and decisions in the future. Deep learning relies on a set of machine learning algorithms that model high-level abstractions in data with multiple nonlinear transformations. Deep learning technology works on an artificial neural network (ANN) system. These neural networks continuously use learning algorithms, and by constantly increasing the amount of data, the efficiency of training processes can be improved. The efficiency of deep learning algorithms depends on the volume of large data. The process is called deep training, because the number of neural network levels increases over time.

The operation in the deep learning process generally depends on two stages called the training and inference.

(i) The training phase involves labeling large amounts of data and determining their adaptive properties.

(ii) The inference step is to conclude and label new and unseen data, using their prior knowledge. Deep learning is a method that helps the system understand the complex tasks of perception with maximum accuracy. Deep learning is also known as deep structured learning and is a hierarchical learning that consists of several layers that include nonlinear processing units to convert and extract features. Each subsequent layer takes the results from the previous layer as input.

The learning process is performed using the distinct stages of abstraction and multiple levels of representation in a supervised or unsupervised manner. Deep learning or deep neural network uses a basic computing unit, a neuron that receives multiple signals as input. It integrates these signals linearly with the weight and transmits the combined signals to the nonlinear tasks to produce output.

In the "deep learning" method, the term "deep" refers to the multiple layers through which data is converted. These systems are composed of a very special deep credit allocation (CAP) path, which means that the steps were performed to convert the input to output and represent the impact connection between the input layer and the output layer [7]. It should be noted that there is a difference between deep learning and machine learning. Machine learning involves a set of methods that help the machine receive raw data as input and set views for the purpose of detection and classification. Deep learning techniques are simply a type of learning method that has several levels of representation and is at a more abstract level. Figure 2 shows the difference between machine learning and deep learning.

Deep learning techniques in large databases use nonlinear transformations and high-level model abstraction. They also describe how a machine can change the internal features needed to count descriptions in each layer by accepting abstractions and displaying previous layers. This new learning approach is widely used in the areas of adaptive testing, big data, cancer diagnosis, data flow, document analysis and identification, healthcare, object recognition, speech recognition, image classification, pedestrian detection, natural language processing, and voice activity detection.

The deep learning model uses a set of features set for large features using bulk dataset for unique features, then extracts a classification model, and creates an integrated classification to explore a variety of applications.

The key factors on which the deep learning method is based are as follows [8]:

(i) Nonlinear processing in multiple layers or stages: nonlinear processing in multiple layers refers to a hierarchical method in which the present layer accepts the results of the previous layer and transmits its output as input to the next layer. Hierarchy is created between layers to organize the importance of the data.

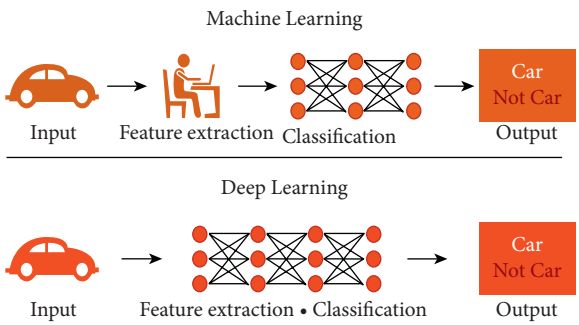


Figure 2: The difference between machine learning and deep learning [8].

(ii) Supervised or unsupervised learning: here, supervised and unsupervised learning are linked to the class goal label. Its availability means a supervised system, and its absence indicates an unattended system.

4. Using Deep Learning to Diagnose Pedestrians

In today's world, where the development of smart cities and smart transportation has received a lot of attention from people, governments, and commercial and manufacturing companies, one of the basic needs is to provide solutions to identify objects around us by sensors and perform appropriate operations according to movements performed by objects. Since in this research we have mainly focused on the development of smart transportation in smart cities, so we will focus only on identifying pedestrians who are influential in the development of smart cars and smart transportation, and studies conducted by various researchers. In the field of pedestrian identification, we have divided these studies into several groups, examining the studies related to each group separately and pointing out the challenges in each.

4.1. Studies Conducted in the Field of Pedestrian Identification in Smart Cities. Belhadi et al. [9] studied the unusual behaviors of pedestrians in smart cities. For this purpose, several algorithms were proposed, which are basically divided into two categories based on performance:

(i) Algorithms that used different data mining and knowledge discovery techniques to discover the relationship between different behaviors of pedestrians, and finally the knowledge generated to identify abnormal behaviors of pedestrians

(ii) Algorithms that have been developed based on the history of pedestrian behaviors and based on different characteristics of the user to detect abnormal pedestrian behaviors

To implement these proposed algorithms, the researchers used the HUMBI dataset (https://humbi-data.net/ (accessed on December 2020)), which contains 164 attributes (including gender, age, and physical condition) that include five basic body parts (including face, hands, body, clothes, and eyes), which were designed using the data in this dataset in this study.

The results of this study showed that the use of deep learning techniques in comparison with the use of data mining techniques both reduces the time of analysis and detection of normal and abnormal behaviors and increases the accuracy of identifying abnormal behaviors of pedestrians. The researchers pointed out that modeling pedestrian behaviors and behavioral analysis in the development of smart cities can help increase the efficiency of smart agents used for various applications in these cities.

Challenge. The limited features used to model pedestrian behaviors and the need to apply metaheuristic algorithms to solve complex intelligent computing problems are among the major challenges in this research to model pedestrian behaviors.

Kim et al. [10] examined pedestrian identification in smart buildings. Because the identification of pedestrians due to noise in images and some environmental factors and parameters faces challenges. The researchers used the Deep Convolution Neural Network (CNN) to create a vision-based model and the optimized version of the VGG-16, called the OVGG-16, as the architectural core used to distinguish pedestrians from the multitude of possible images. To evaluate the proposed method, the researchers used the INRIA Dataset (http://pascal.inrialpes.fr/data/human/ (accessed on December 2020)), which contained 6817 images with 3239 pedestrian images, and the image quality in this dataset was 227×227 pixels. The results of the researchers' studies showed that the proposed method has a high accuracy (approximately 98.8%) for the correct identification of pedestrians compared to other methods of machine learning.

Challenge. The model created on noisy data has not been evaluated, and there is a question: if the set of images and input data has a lot of noise, how accurate will the pedestrian be identified by this proposed model?

Using deep learning, Tomè et al. [11] proposed a system for pedestrian identification. The researchers also proposed a new framework for identifying pedestrians. The researchers also proposed new solutions for different stages of pedestrian detection, which used deep learning to easily implement their proposed algorithm on modern hardware. To implement and evaluate the proposed methods and solutions, they used the NVIDIA Jetson TK1, a GPU-based computing platform (https://developer.nvidia.com/embedded/jetson-tk1-developer-kit (accessed on December 2020)), and the Caltech Pedestrian dataset (http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/ (accessed on December 2020)). This dataset contains about 10 hours of video content related to vehicles collected in different weather conditions. This dataset had 250 k frames per 137 minutes of video content with 2300 different pedestrians. Half of the frames had no pedestrians, and 30% of the frames had 2 or 3 pedestrians. The results of the implementation of these researchers showed that their proposed method has high efficiency and accuracy in identifying pedestrians in real time.

Challenge. To implement these methods, we need large amounts of data, and the more data the dataset uses, the more efficient and accurate the proposed method will be. The challenge arises when data collection for various reasons (including privacy) may not be possible in the metropolitan areas of many countries in high-traffic urban areas.

4.2. Studies Conducted in the Field of Pedestrian Identification for the Development of Intelligent Transportation Systems and Self-Driving Cars. Chen et al. [12] examined existing architectures for pedestrian detection when using the automated driving method. These researchers first explained the need to use methods to identify a pedestrian and determine his or her route and then discussed the process of identifying a pedestrian while driving a car. They, then, discussed how to use deep learning techniques (such as R-CNN, SVM) to discover two-step and one-step patterns and test the effectiveness of the patterns discovered to identify pedestrians. Finally, the researchers examined and compared methods proposed by other researchers to identify pedestrians. They also introduced several datasets (such as KTH, the UCF series, Hollywood2, and Google AVA) that are used to examine proposed methods for detecting pedestrian movement.

Challenges. In this research, several important challenges in identifying pedestrians are mentioned, which are as follows:

(i) The complexity of the environment around the pedestrian can overshadow the operations and methods of recognizing the pedestrian and his movement and, as a result, make it difficult to accurately identify the pedestrian. Therefore, creating methods to identify different perspectives on pedestrian detection and operations performed by him is one of the challenges mentioned in this research.

(ii) Pedestrian coverage can be extremely effective in the process of identifying him/her. If the images are taken from one perspective, this can affect the accuracy of pedestrian identification and reduce the accuracy of identification. Therefore, it is necessary for researchers to propose new methods for preparing multidimensional images and their simultaneous study and aggregation of the results for early identification of pedestrians, especially in self-driving cars.

(iii) At present, there is no standard for determining the operations and actions in a vehicle against various movements performed by pedestrians. Better results can be obtained from the effects of identifying pedestrians (such as monitoring the safety of passengers and the driver while traveling and managing environmental pollution in cities) by creating a classification and stating more details about driving practices in self-driving cars.

Said and Barr [13] proposed a new program using deep learning algorithm for fast and accurate pedestrian detection to provide real-time responses in driver assistance systems. This program uses object classification and pedestrian identification and location tracking. The TensorFlow deep learning framework, Nvidia, cuDNN, and OpenCv acceleration libraries, and the Caltech dataset were used to implement, learn, and test the proposed method. This program is installed for deployment in mobile phones or Embedded Systems connected to self-driving cars in order to develop driver assistance systems.

Challenge. Real-time and accurate detection of objects (such as pedestrians) is one of the major challenges in the automotive industry to create and develop self-driving cars. With all the efforts that have been made, the percentage of pedestrian detection accuracy and the speed of detection of existing methods are not enough, so these methods are not very acceptable for applying real-time responses.

Ahmed et al. [14] first compared the methods and techniques used to diagnose pedestrians and cyclists. They stated that because of, in the detection stage, the possibility of detecting and locating objects (using deep learning techniques such as fast region-convolutional neural network (R-CNN), faster R-CNN, and single shot detector (SSD)) in images and video frames, so the detection stage can be created as a vital part in creating and developing smart applications in a self-driving vehicle. Finally, tracking results can be used to monitor and identify pedestrians or cyclists. The main purpose of this study was to investigate the existing methods for identifying cyclists. The results of their studies showed that the use of appropriate techniques (e.g., sensor fusion and intent estimation) for identifying pedestrians and cyclists can be an important step in maintaining road safety. In this research, first, the challenges in identifying and estimating the purpose and destination are presented, then a history of methods proposed by various researchers for pedestrian detection is presented, and the general steps proposed for object detection are explained. Next, the research conducted by other researchers on the use of deep learning techniques and architectures to identify pedestrians is reviewed, and then, the dataset used by various researchers to implement their proposed methods for pedestrian and cyclist detection is explained.

Challenge. Most of the existing datasets for implementing object detection techniques are focused on pedestrian detection data, and there is no dedicated dataset to implement the proposed techniques for identifying cyclists, so collecting this type of dataset in different areas is currently a challenge.

Zhu et al. [15] studied the challenges of pedestrian detection using infrared and proposed to use deep learning methods for pedestrian detection to overcome these challenges. By combining deep learning and background subtraction methods, the researchers proposed a new method for pedestrian detection. The proposed algorithm had two steps for pedestrian detection, which are as follows:

Step 1: background subtraction methods are performed to provide information between frames for the machine learning module

Step 2: refine Det equipment with a module of attention that is used to improve the accuracy of identifying pedestrians who are small in stature

In this study, a dataset consisting of infrared videos was created that was used to identify pedestrians from a distance and had good performance.

Challenge. The proposed method in this research is based on a dataset created by the researcher, and its performance is well evaluated. This is a set of video data stored by infrared. In order to prove the effectiveness of this method, it seems that it is necessary to use other datasets that have been collected from different geographical locations with different volumes of pedestrian traffic.

Bunel et al. [16] focused on remote pedestrian detection. When the pedestrian is too far from the camera, the size of the pedestrian becomes very small, so it becomes very difficult to detect. These researchers suggested a neural network-based method and convolutional neural network-based method to learn the features with an end-to-end approach to identify pedestrians who are too far away from the camera and too visible. Further, in this research, to implement the proposed method, they used Caltech Pedestrian Datasets. The results showed that the proposed method has a good performance in identifying pedestrians.

Challenge. It seems that the quality of cameras and the amount of pedestrian distance from the camera can be effective in the performance of the proposed method and the accuracy of method detection, which has not been considered in this study. Therefore, it is recommended to accurately define a standard for the best image quality and accuracy of diagnoses with different measurements and different photographs or videos.

Haghighat et al. [17] investigated the application of deep learning models in intelligent transportation systems. In the following, the advantages and disadvantages of embedded systems were discussed, and finally, the use of deep learning techniques to predict the occurrence of traffic on different road routes was examined.

Challenge. All datasets used in research conducted by different researchers have been collected using cameras installed in different areas. It is expected that, with the advances made in the production of self-driving cars and sensors used on cars or on the street floor, the volume of data collected from them will be greatly increased, so we need new techniques in deep learning to be able to analyze this data.

Yu et al. [18] proposed a system for tracking and identifying pedestrians using deep neural networks, which used a UAV and Kalman Filter forecasting method to track objects and pedestrians, and a dataset (YOLOv3) was used to implement the proposed method. To measure the efficiency of the proposed method for tracking and identifying pedestrians, accuracy and execution time and observing and identifying objects were examined. The results of experimental experiments showed that the proposed method had fewer errors in identifying pedestrians.

Challenge. The dataset used in this study had a very small number of records. Therefore, it seems that a larger dataset can be used to get better results. Also, this method should be tested to identify other objects (such as cars and cyclists), and its accuracy should be checked to identify those objects.

Dinakaran et al. [19] proposed generative adversarial networks (GANs) to create a new Cascaded Single Shot Detector (SSD) architecture for remote pedestrian detection. In this architecture, DCGAN is used to improve the image quality for remote pedestrian detection. In this proposed method, several criteria are used to identify the objects in the image. To implement the proposed method, the dataset of the Canadian Institute for Advanced Research (CIFAR) is used. The results obtained from experiments have shown that the proposed method has a high accuracy in identifying vehicles and pedestrians from a distance.

Challenge. Generative adversarial networks (GANs) can be used to remotely detect objects in smart cities. Since security in communications created in IoT-connected networks in smart cities is very important and fundamental, so we conduct research on the use of GANs in improving security in smart cities to identify vehicles and pedestrians. Immediately, it seems very necessary.

To overcome the problem of Occlusion handling, Tian et al. [20] proposed DeepParts, which consists of extensive trackers, instead of using deep learning techniques with an image detector. Some of the features of DeepParts are as follows:

(i) First, these DeepParts can be trained with poorly labeled data

(ii) Second, DeepParts is able to handle low IoU positive proposals that shift away from ground truth

(iii) Third, every part detector in DeepParts is a powerful tracker that can detect a pedestrian by observing only a part of the body

To implement the proposed method in this research, Caltech and KITTI datasets were used, and its performance is compared to other detectors used for pedestrian detection.

Challenge. It is expected that, by using the combination of the results obtained from all the detectors used in the parts, the accuracy of identifying objects, especially pedestrians, will be increased. The use of deep learning techniques and other techniques on data from detectors may improve pedestrian detection accuracy.

In Navarro Lorente et al. [21], an automated sensor-based system was used in applications on self-driving vehicles to identify pedestrians. Different types of sensors are used in self-propelled vehicles, but in this study, researchers focused on the Velodyne HDL-64E LIDAR sensor. The data generated by this sensor was analyzed in three dimensions using machine vision and machine learning algorithms (such as nearest neighbor algorithm, Bayesian classification and support vector machine). A new framework called the Renault Twizy platform was proposed in this study to develop the ability of self-driving vehicles to identify pedestrians. The results of the implementation of their proposed framework showed that their selected features along with the

algorithms used and the quality of the camera can be important factors in better identifying pedestrians and motorcyclists.

Challenge. Implementing the algorithms used is time-consuming, and it is necessary to propose methods for accurate and real-time identification of pedestrians and motorcyclists.

Combs et al. [22] focused on using sensors installed on self-driving cars to reduce the number of deaths due to human-caused traffic accidents. They used the Fatality Analysis Reporting System (FARS) to track the number of human error deaths on US urban and suburban roads. The researchers hypothesized that a car was traveling on a road and had all the necessary sensors to detect a pedestrian and fully effective software to detect and analyze the movements and movements made by a pedestrian to identify that pedestrian. In addition, sensors mounted on the vehicle itself are able to receive signals from the movement of pedestrians. As a result, a model can be developed to be able to easily identify pedestrians and prevent accidents. The proposed model used data from VLC cameras, radar-based detection systems, and light amplitude detection (LiDAR). The results of their practical tests showed that, by using these facilities along with sensors installed on the car body, 90% of accidents caused by human error can be prevented, while using only VLC can reduce the accident statistics by only 30%.

Challenge. There is high cost of using sensors, VLC. LiDAR and car-based radar detection systems prevent automakers from using all of them to prevent rising car prices, or from using low-quality cameras, which could be a reason to reduce its quality and ultimately reduce the accuracy of identifying pedestrians or obstacles on the road.

Song et al. [23] proposed an algorithm for detecting pedestrians on the road. In this study, the pedestrian target area and the results of pedestrian detection on the road by combining the algorithm most similar to the neighbor and the least energy algorithm were accurately divided. In this study, all objects that are around a car and can generate traffic for pedestrian identification (such as cyclists, trees, other cars, and buildings around cars) are divided. And then, an algorithm was proposed for the environmental coverage of the road. The researchers used several experiments to evaluate the performance of the traffic-generating object classification detection system proposed in this study. They selected and examined several sequences of images, including different road scenes, different weather conditions, and different city streets.

Challenge. The time required to identify each of the obstacles (especially pedestrians) is variable, but in changing weather conditions and different road conditions, the time to identify obstacles and pedestrians can be increased or decreased. Therefore, we need to optimize the proposed algorithm in this research to realize response time and identify pedestrians.

Hbaieb et al. [24] proposed a new method for detecting the presence of pedestrians in the path of self-driving cars through an intervehicle communication system. In this method, descriptor (HOG), support vector machine classification (SVM), pedestrian tracker, and feature-based cascade classification were used to achieve vehicle detection. The performance evaluation results of the proposed method were about 90% for pedestrians and 88% for vehicle detection.

4.3. Proposed Methods for Pedestrian Detection Using Different Techniques. Cai et al. [25], to solve the problems caused by resizing objects at the accuracy and speed of object identification, provide a deep unified neural network, representing the multicast CNN (MS-CNN), for the rapid detection of multifunctional objects. They gave MS-CNN including a proposed subnet and an identification subnet. The proposed subnet has several output layers, in which objects are detected at different scales. The detection subnet uses tracking methods for multipurpose object monitoring. The proposed method was implemented on the KITTI and Caltech datasets, and the results showed that the proposed method has a very good performance in detecting objects with a maximum of 15 frames per second.

Challenge. In this research, the CNN feature approximation has been used as an alternative to input sampling. The challenge is whether other methods can be used to sample the inputs that save more memory and time for calculations. Is it possible to increase the speed of moving and replacing frames (above 15 frames per second)?

Fukui et al. [26] used complex neural network-based (CNN) methods that are highly accurate in a variety of contexts to identify pedestrians. The researchers proposed a new method proposed in this research based on CNN and used Random Dropout and Ensemble Inference Network (EIN) for training and classification, respectively. Random Dropout selects units that have a high and variable flexibility rate for training, while, in a typical dropout, the flexibility rate is fixed. EIN creates multiple networks with different structures in well-connected layers. The researchers used the Caltech and Daimler Mono pedestrian datasets to implement their proposed method.

Challenge. The costs of real-time calculations to identify pedestrians using this proposed method are relatively high, so it is necessary to adopt methods to reduce these costs.

To achieve better performance in applying deep learning theory to pedestrian detection, Cai et al. [27] improved the performance of a poorly supervised hierarchical deep learning algorithm with two-dimensional deep belief networks. In the proposed design of this research, the weaknesses of the structure and training methods used in the various algorithms of the existing classifications are identified, and the following operations are performed to eliminate these weaknesses:

(i) First, a network of deep one-dimensional beliefs expands to two-dimensional, allowing the image matrix to be loaded directly to preserve more information from the sample space.

(ii) Second, a lightweight regulation term is added to performance consistent with the training goal

without the use of traditional oversight. With this reform, the main training without supervision becomes weak training under supervision.

(iii) Third, the ability to distinguish between these extracted features is created.

(iv) In this research, the INRIA, Daimler, and CVC datasets of Spain have been used to implement and evaluate the accuracy of the proposed method.

Challenge. Working with unstructured data with existing traditional methods faces several challenges (including challenges in the preprocessing, analysis, and grouping stages). It is necessary to adopt methods or algorithms to optimize the performance of existing methods for analyzing that data and identifying pedestrians through the results of those analyzes. It is also necessary to adopt strategies to improve the performance of algorithms and methods used to classify data and semantic information in occlusion conditions.

Saeidi and Ahmadi [28] first examined some of the DCNN-based learning methods and briefly explained the new algorithms proposed by various researchers for these methods. Next, the researchers proposed a deep architectural method and a new training method based on parallel DCNNs for pedestrian detection. The proposed method had two stages of training, which are as follows:

(i) Learning Candidate Pedestrian Extractor Network (CPEN) Candidate for pedestrian training

(ii) Parallel training DCNNs (PDCNNs) to teach how to identify a candidate pedestrian by identifying the body parts of that candidate pedestrian

In this study, the Caltech-USA dataset was used to implement the proposed method. The results obtained from evaluating the accuracy of the proposed method in pedestrian detection and comparing it with other methods showed that this method has a higher accuracy compared to other methods.

Challenge. Selecting features for pedestrian detection, especially in multidimensional data, is one of the most important challenges when using deep learning techniques. Deep learning techniques (such as SquaresChnFtrs, InformedHaar, and Katamari) performed poorly in selecting effective features for pedestrian detection, but deep learning techniques have recently been proposed by various researchers (e.g., CompAct-Deep [29], DeepParts [20], and TA-CNN [30]). They performed much better in selecting suitable features for pedestrian detection.

Vasconcelos et al. [31] proposed an automated method for optimizing the efficiency of the training suite by creating deformation and creating a local perspective. Using this method, human figures can be identified in the existing training set by applying monitoring scenarios. Experimental results of applying this method to datasets that included a variety of data and images (selection of 16 features from the imageNet dataset [32]) showed that if these data were entered as input to a convolutional neural network, it will be able to identify pedestrians with high accuracy. This rich image database can be used in other detectors based on supervised learning architecture.

Challenge. Creating datasets with a number of effective features for pedestrian detection is one of the important challenges that the more datasets we use have a variety of effective features for pedestrian detection and can more accurately identify pedestrians, using the method proposed in this study.

Zeng et al. [33] first focused on in-depth collective public learning about each of the factors used to identify pedestrians using advances in creating a new deep neural network architecture. The proposed architecture in this research has the following parts:

(i) Filtered information maps are obtained from the first convolution layer.

(ii) From the second convolution layer, maps are obtained to identify parts of the image.

(iii) The results obtained by identifying each part of the pedestrian body are used to track maps and work with information obtained from layers. Argument about access to 20 feature parts or parts of the pedestrian body is used to estimate the tag (for example: does a particular window have a pedestrian or not?).

(iv) The windows are provided in dimensions (height 84 and width 28) that the dimensions of the pedestrian can be identified by 60 by 20.

In other words, the proposed method in this research has four parts for pedestrian detection, which are feature extraction, handling deformation, handling of occasions, and classification.

The proposed method and architecture were implemented using Caltech and ETH datasets, and their efficiency and accuracy in pedestrian identification were compared with the accuracy of other deep learning methods. The results show that the accuracy of the proposed method in this research is higher than that of other methods.

Challenge. To extract the effective features in high-precision pedestrian detection, we need a large dataset with a large number of features that were not available in this study; so, to ensure the accuracy of the proposed method in this study, we need to prepare a very large dataset with more features.

Tarchoun et al. [34] proposed two methods for tracking pedestrians in images taken from moving vehicles:

(i) In the first method, the block matching algorithm and block matching features are used to identify pedestrians

(ii) The second method uses a faster R-CNN detector to detect pedestrians

The proposed methods were implemented using the I2V-MVPD database, and the results showed that the first method was able to detect pedestrians in images obtained from moving vehicles in less time but had a higher false

positive rate compared to the second method. The second method had better accuracy and performance in pedestrian detection.

Challenge. Neither of these two methods can be used for real-time pedestrian detection applications, so more research is needed to reduce costs and time on these two methods.

Lee et al. [35] proposed a deep fusion network-based pedestrian detection method that used a single shot multibox detector (DSSD) halfway through. They use correlations between other feature maps to create new properties. In this study, deep fusion network was used to form issues related to the method of recognizing color images at night or pedestrian images in the dark. KAIST dataset was used to implement the proposed method. The results obtained from the implementation and evaluation of the results showed that the proposed method, compared to other methods, had at least 4.28% lower error rate in identifying pedestrians in the dark environments.

Challenge. Correctly identifying and exacting location of pedestrians in the dark using existing methods is still a challenge. Creating ways to connect different features and deep learning techniques can go a long way in increasing the accuracy of identifying pedestrians in the dark.

Ribeiro et al. [36] proposed a deep learning method for pedestrian detection (PD) detection in real time to solve problems related to the human-aware robot navigation problem. To achieve fast and accurate pedestrian detection efficiency, this study developed a combination of Aggregate Channel Features (ACF) detector with a deep convolutional neural network (CNN). In this method, we have tried to use CNN to increase the accuracy of pedestrian detection by trackers. To implement the proposed method and evaluate its accuracy, two sets (called corridor and Mbot) were used, which have real photos taken by the cameras (photos collected from the cameras in the internal and external sensors of the robot), and a typical robot navigation environment was used to evaluate the accuracy of the method in identifying pedestrians, and the results showed that it has sufficient speed and accuracy to be used in these environments and robot navigation applications to identify pedestrians.

Challenge. The performance of the proposed method should be evaluated on datasets collected from cameras located in different places with different light intensities and distances, different types of sensors installed in the environment such as laser sensors.

Hu et al. [37] worked to create a powerful pedestrian detector. For this purpose, the researchers designed the deep convolutional neural network (DCNN) as an image feature to teach a set of enhanced decision models, using redesigned learning algorithms (CFMs) without the use of learning algorithms. To increase the efficiency and accuracy of DCNN-based detectors for image detection of pedestrians, hand-crafted features such as optical flow are used. In this study, they reviewed various datasets that have been used by other researchers to implement their proposed methods for pedestrian detection. They used the KITTI dataset to implement the proposed method in this research. The results of the evaluation of the proposed method showed that this method reduces the complexity of the detectors and can be more efficient in accurately identifying pedestrians.

Challenge. The proposed method in terms of time required to identify pedestrians may be associated with challenges; i.e., in terms of time, more studies should be done on this method, so that it can be used immediately to detect pedestrians in cars used.

Wagner et al. [38] explored the potential of deep learning techniques in pedestrian identification. They examined two deep fusion architectures and their performance on multispectral data. Finally, they used a new deep CNN-based method to detect pedestrians based on multispectral image data to analyze the proposed method. They introduced the first deep CNN application for pedestrian detection based on multispectral image data, and they used three datasets (including ImageNet [32], CALTECH benchmark [2], and KAIST) to implement and evaluate the proposed method. The evaluation results showed that the proposed method had a higher accuracy in pedestrian detection compared to other methods.

Challenge. The most important challenge in the proposed method is that, most of the time, early-fusion architecture is not able to achieve our expected performance. The reason for this may be due to the inability of the early-fusion network to learn the meaningful multistate abstract properties in a given environment.

Kim et al. [39] proposed a system with limited resources for real-world monitoring and identification of moving persons. For this purpose of combination background subtraction and convolutional neural networks (CNNs), they used it to identify and detect moving objects using outdoor CCTV videos. The background subtraction algorithm used to find the desired areas in the video frame and the CNN classifier was used to classify the ROIs obtained in one of the predefined classes. To implement the proposed method in practice, various datasets collected by several real-world CCTV cameras were used. The results showed that the proposed system had a high accuracy in identifying pedestrians and was also less complex than other methods.

Challenge. Occurrence of some problems in the collected data can reduce the performance or accuracy of pedestrian detection. For example, lack of training data may disrupt the training process. On the other hand, using the same images will cause a pedestrian to be repeatedly identified several times, and this will reduce the performance of the proposed system for pedestrian detection.

Lin et al. [40] proposed a framework for pedestrian detection that is based on incorporating pixel-wise information into deep convolutional feature maps. In this context, they used the zooming properties to improve image quality to help easily and accurately identify pedestrians. Therefore, the proposed method in this research helps

Table 1: Challenges and proposed solutions.

Challenges	Proposed solutions
Lack of structured data in all studied routes to identify pedestrians at different times of the day with different light intensities during the day and different weather conditions	Installation of devices in the desired routes to monitor the passage of pedestrians and bicycles around the clock and the use of various data mining techniques and deep learning to create patterns to identify cyclists and pedestrians in different weather conditions and different light intensities
Absence of any known standard for selecting appropriate and essential features from the features collected for pedestrian identification or limited features used to model pedestrian behaviors and the need to apply metaheuristic algorithms to solve complex intelligent computing problems	Using more advanced sensors in different directions or applying sensors in the car body, creating a centralized database to consolidate data collected from sources and sensors used in different places
Existence of noise in various images and data collected, poor quality or very low quality of some images collected by cameras installed on the road, especially in cloudy, rainy and icy weather or dark at night	Creating high-precision and high-quality sensors and cameras, installing high-quality cameras on sensitive routes, proposing new techniques for preprocessing data and images, and accurately detecting noise images
Lack of access to clear and uninterrupted data and images for research due to privacy	Establish protocols and standards for collecting data on public places, establish agreements and laws to protect the privacy of the public while collecting 24-hour data from all road routes
The complexity of the environment around the pedestrian can overshadow the operations and methods of identifying the pedestrian and her movement, and as a result, make it difficult to accurately identify the pedestrian	Develop methods to identify different perspectives on pedestrian detection and operations performed by his/her
Pedestrian coverage can be very effective in the process of identifying him. If the images are taken from one perspective, it can affect the accuracy of pedestrian identification and reduce the accuracy of identification	Proposing new methods by researchers to prepare multidimensional images and their simultaneous study and aggregation of the results for early identification of pedestrians, especially in cars
Real-time and accurate detection of objects (such as pedestrians) is an important challenge for car companies to create and develop self-driving cars	Propose methods to increase the accuracy of pedestrian detection and speed up the process of immediate response to avoid accidents when detecting obstacles and pedestrians
Lack of specific data to diagnose cyclists	Collect data on cyclists on different routes with different light intensities and create patterns to identify cyclists
The quality of the cameras, the amount of pedestrian distance from the camera can affect the performance of the proposed method and the accuracy of the method detection, which has not been considered in many studies	Define a standard for the best image quality and detection accuracy with different measurements and different shots or videos
The cost of performing real-time calculations to identify pedestrians using this proposed method is relatively high	It is necessary to adopt methods to reduce these costs
Working with unstructured data with existing traditional methods faces several challenges (including the challenges of preprocessing, analysis, and grouping)	It is necessary to adopt methods or algorithms to optimize the performance of existing methods for analyzing that data and identifying pedestrians through the results of those analyzes. It is also necessary to adopt strategies to improve the performance of algorithms and methods used to classify data and semantic information in occlusion conditions

identify pedestrians who are seen in a very small image by inserting geographical location specifications and pedestrian features. The proposed method uses three datasets: Caltech [41], INRIA [42], and KITTI [43]. The implementation results obtained from the evaluation and comparison with other methods showed that this method is more efficient in terms of reducing the time of pedestrian identification and the number of unidentified cases.

Challenge. Due to the small size of the pedestrian image, it seems that there are complications in recognizing of that pedestrian in an image taken from a low light environment, especially at night, using the method proposed in this research.

Dollár et al. [41] reviewed advances over the past decade in developing methods for pedestrian detection and proposing 40 trackers for pedestrian detection. They analyzed the performance of these detectors using various datasets including Caltech. In this study, the most widely used datasets were briefly described, and the strengths and weaknesses of each were expressed. Three features (including best features, additional data, and background/conceptual information) were used to conduct practical experiments in this study, which can affect the efficiency of the proposed method for pedestrian detection. Three important and famous trackers (including deformable part models, decision forests, and deep networks) are based on the different learning techniques used.

Challenge. It seems that the most important challenge in the field of pedestrian detection is to develop a deeper understanding in selecting the best features to achieve the highest accuracy and performance in real-time pedestrian detection.

The main challenge ahead seems to develop a deeper understanding of what makes good features, so as to enable the design of even better ones.

5. Conclusion

Nowadays, the amount of data generated during a day from various sensors and other devices is enormous, and technologies such as cloud computing are being used to help store large volumes of collected data. One of the benefits of analyzing this data is the discovery of knowledge and a pattern for use in similar situations by training machines. In this study, we explained deep learning and its difference with machine learning, and then some research is done by various researchers around the use of deep learning techniques in the creation and development of smart cities and pedestrian identification in smart cities and Intelligent Transportation Systems (ITS). Finally, we examined smart transportation and listed the challenges in each of them.

In general, according to the studies conducted in this paper, the most important challenges in identifying pedestrians on the street using the proposed technologies and methods, especially deep learning techniques, can be expressed in Table 1 and some of the solutions that seem useful in solving these challenges are suggested in this table.

Data Availability

This is a survey, and we introduced already all databases in this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. E. Schapire and Y. Freund, *Foundations of Machine Learning*, MIT Press, Cambridge, MA, USA, 2012.

[2] ETSI, "Intelligent transport systems (ITS); security; threat, vulnerability and risk analysis (TVRA)," European Telecommunications Standards, Sophia Antipolis, France, ETSI TR 102 893, 2010.

[3] T. Rawal and V. Devadas, "Intelligent transportation system in India—a review," *Journal of Management Development*, vol. 2, p. 299, 2015.

[4] P. Fernandes and U. Nunes, "Vehicle communications: a short survey," in *Proceedings of the IADIS Telecommunications, Networks and Systems*, p. 1556, Lisboa, Portugal, July 2007.

[5] K. B. Smith, "Typologies, taxonomies, and the benefits of policy classification," *Policy Studies Journal*, vol. 30, no. 3, pp. 379–395, 2002.

[6] T. Brosch, R. Tam, and Initiative for the Alzheimers Disease Neuroimaging, "Manifold learning of brain MRIs by deep learning," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, Nagoya, Japan, September 2013.

[7] R. Heffernan, K. Paliwal, J. Lyons et al., "Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning," *Scientific Reports*, vol. 5, no. 1, p. 11476, 2015.

[8] S. Dargan and M. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," *Archives of Computational Methods in Engineering*, vol. 27, pp. 1071–1092, 2019.

[9] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, and G. Fortino, "Deep learning for pedestrian collective behavior analysis in smart cities: a model of group trajectory outlier detection," *Information Fusion*, vol. 65, pp. 13–20, 2021.

[10] B. Kim, N. Yuvaraj, K. R. Sri Preethaa, R. Santhosh, and A. Sabari, "Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance," *Soft Computing*, vol. 24, no. 22, pp. 17081–17092, 2020.

[11] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, "Deep convolutional neural networks for pedestrian detection," *Signal Processing: Image Communication*, vol. 47, pp. 482–489, 2016.

[12] L. Chen, N. Ma, P. Wang et al., "Survey of pedestrian action recognition techniques for autonomous driving," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 458–470, 2020.

[13] Y. F. Said and M. Barr, "Pedestrian detection for advanced driver assistance systems using deep learning algorithms," *IJCSNS (International Journal of Computer Science and Network Security)*, vol. 19, no. 10, 2019.

[14] S. Ahmed, M. N. Huda, S. Rajbhandari, C. Saha, M. Elshaw, and S. Kanarachos, "Pedestrian and cyclist detection and intent estimation for autonomous vehicles: a survey," *Applied Sciences*, vol. 9, no. 11, p. 2335, 2019.

[15] Y. Zhu, J. Yang, X. Xieg, Z. Wang, and X. Deng, "Long-distanceinfrared video pedestrian detection using deep learning and backgroundsubtraction," *Journal of Physics: Conference Series*, vol. 1682, 2020.

[16] R. Bunel, F. Davoine, and P. Xu, "Detection of pedestrians at far distance," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Stockholm, Sweden, May 2016.

[17] A. K. Haghighat, V. Ravichandra-Mouli, P. Chakraborty, Y. Esfandiari, S. Arabi, and A. Sharma, "Applications of deep learning in intelligent transportation systems," *Journal of Big Data Analytics in Transportation*, vol. 2, no. 2, pp. 115–145, 2020.

[18] H. Yu, G. Li, W. Zhang et al., "The unmanned aerial vehicle benchmark: object detection, tracking and baseline," *International Journal of Computer Vision*, vol. 128, no. 5, pp. 1141–1159, 2020.

[19] R. K. Dinakaran, P. Easom-McCaldin, A. Bouridane et al., "Deep learning based pedestrian detection at distance in smart cities," in *Intelligent Systems and Applications*, Springer, Cham, Germany, 2019.

[20] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.

[21] P. J. Navarro Lorente, C. Fernandez, R. Borraz, and D. Alonso, "A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3D range data," *Sensors*, vol. 17, no. 1, p. 18, 2017.

[22] T. S. Combs, L. S. Sandt, M. P. Clamann, and N. C. McDonald, "Automated vehicles and pedestrian safety: exploring the promise and limits of pedestrian detection," *American Journal of Preventive Medicine*, vol. 56, no. 1, pp. 1–7, 2019.

[23] Y. Song, J. Yao, Y. Ju, Y. Jiang, and K. Du, "Automatic detection and classification of road, car, and pedestrian using binocular cameras in traffic scenes with a common framework," *Complexity*, vol. 2020, Article ID 2435793, 17 pages, 2020.

[24] A. Hbaieb, J. Rezgui, and L. Chaari, "Pedestrian detection for autonomous driving within cooperative communication system," in *Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, Marrakesh, Morocco, April 2019.

[25] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*, Springer, Cham, Germany, 2016.

[26] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, "Pedestrian detection based on deep convolutional neural network with ensemble inference network," in *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, Seoul, South Korea, July 2015.

[27] Y. Cai, Y. He, H. Wang, X. Sun, L. Chen, and H. Jiang, "Pedestrian detection algorithm in traffic scene based on weakly supervised hierarchical deep model," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, Article ID 1729881417692311, 2016.

[28] M. Saeidi and A. Ahmadi, "Deep learning based on parallel CNNs for pedestrian detection," *International Journal of Information and Communication Technology Research*, vol. 10, no. 4, pp. 42–52, 2018.

[29] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, December 2015.

[30] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, June 2015.

[31] C. N. Vasconcelos, A. Paes, and A. Montenegro, "Towards deep learning invariant pedestrian detection by data enrichment," in *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Anaheim, CA, USA, December 2016.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, June 2009.

[33] X. Zeng, W. Ouyang, and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Sydney, Australia, December 2013.

[34] B. Tarchoun, A. Khalifa, S. Dhifallah, I. Jegham, and M. Mahjou, "Hand-crafted features vs deep learning for pedestrian detection in moving camera," *Traitement du Signal*, vol. 37, no. 2, pp. 209–216, 2020.

[35] Y. Lee, T. D. Bui, and J. Shin, "Pedestrian detection based on deep fusion network using feature correlation," in *Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, Honolulu, HI, USA, November 2018.

[36] D. Ribeiro, R. Mateus, P. Miraldo, and J. C. Nascimento, "A real-time deep learning pedestrian detector for robot navigation," in *Proceedings of the 2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, Coimbra, Portugal, April 2017.

[37] Q. Hu, P. Wang, C. Shen, A. v. d. Hengel, and F. Porikli, "Pushing the limits of deep CNNs for pedestrian detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1358–1368, 2017.

[38] J. Wagner, V. Fischer, M. Herman, and S. Behnke, "Multispectral pedestrian detection using deep fusion convolutional neural networks," in *Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, April 2016.

[39] C. Kim, J. Lee, T. Kan, and Y.-M. Kim, "A hybrid framework combining background subtraction and deep neural networks for rapid person detection," *Journal of Big Data*, vol. 5, no. 1, pp. 1–24, 2018.

[40] C. Lin, J. Lu, G. Wang, and J. Zhou, "Graininess-aware deep feature learning for pedestrian detection," in *European Conference on Computer Vision (ECCV)*, Springer, Cham, Germany, 2018.

[41] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: a benchmark," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009.

[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, San Diego, CA, USA, June 2005.

[43] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Providence, RI, USA, June 2012.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3790888, 15 pages
https://doi.org/10.1155/2021/3790888

Research Article

Rebalancing Strategy for Bike-Sharing Systems Based on the Model of Level of Detail

Zhenghua Hu,[1,2] **Kejie Huang**,[1] **Enyou Zhang**,[3] **Qi'ang Ge**,[2] **and Xiaoxue Yang**[2]

[1]*College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou, China*
[2]*School of Electronic and Information Engineering, Ningbo University of Technology, Ningbo, China*
[3]*Ningbo Jianan Electronics Co., Ltd, Cixi, China*

Correspondence should be addressed to Kejie Huang; huangkejie@zju.edu.cn

Received 2 July 2021; Revised 10 August 2021; Accepted 14 September 2021; Published 27 September 2021

Academic Editor: Chi-Hua Chen

Traveling by bike-sharing systems has become an indispensable means of transportation in our daily lives because green commuting has gradually become a consensus and conscious action. However, the problem of "difficult to rent or to return a bike" has gradually become an issue in operating the bike-sharing system. Moreover, scientific and systematic schemes that can efficiently complete the task of rebalancing bike-sharing systems are lacking. This study aims to introduce the basic idea of the *k*-divisive hierarchical clustering algorithm. A rebalancing strategy based on the model of level of detail in combination with genetic algorithm was proposed. Data were collected from the bike-sharing system in Ningbo. Results showed that the proposed algorithm could alleviate the problem of the uneven distribution of the demand for renting or returning bikes and effectively improve the service from the bike-sharing system. Compared with the traditional method, this algorithm helps reduce the effective time for rebalancing bike-sharing systems by 28.3%. Therefore, it is an effective rebalancing scheme.

1. Introduction

The issue of traffic congestion is becoming increasingly serious with the increase in the number of motor vehicles in urban areas [1]. This situation has become a major problem in the development of modern cities. In many big cities in China, the running speed of cars on the arterial roads during rush hours is less than 20 km/h. Persistent traffic congestion and a series of social problems, such as environmental pollution, energy waste, and exhaust emissions, have seriously hindered the development of economy. Nevertheless, people are paying increasing attention to environmental protection, energy saving, and emission reduction because of the increasing transportation problems and the deteriorating ecological environment. The concept of green commuting has also attracted increasing attention, and the vigorous development of a public transport system in cities has become the community consensus. Traveling by bike-sharing system is regarded as a green and low-carbon way of transportation [2]. This mode of transport can alleviate the traffic pressure on the road and solve "last mile" problem in public transportation [3]. Therefore, traveling by bike-sharing system has become increasingly popular and received the attention of governments worldwide [4–7].

However, bikes cannot spontaneously form a relatively balanced distribution due to the asymmetry of traffic flow and the uncertainty of needs from users. The government has utilized a considerable proportion of manpower and financial resources to rebalance bikes. However, situations such as "no bike to rent" or "no pile to return a bike" still exist in many stations due to the hysteresis and non-scientificity of the rebalance [8, 9]. Consequently, many citizens abandon bike-sharing system when they go out, thereby greatly reducing the probabilities for people to use bike-sharing system [10–12] and restricting the development of bike-sharing systems to a certain extent.

Simply increasing facilities to improve the service of the bike-sharing system and increase the satisfaction from the public cannot fundamentally solve the problem. How to effectively rebalance bikes among stations to ensure that

stations with fewer bikes can be quickly transferred in, whereas those with more bikes can be transferred out, has become a topic of common concern for more and more researchers [13, 14].

This study develops a rebalancing scheme based on the model of level of detail and the hierarchy of cognitive processes. This method performs the rebalancing scheme with granularity from coarse to fine throughout the whole experimental area. By taking the bike-sharing system in Ningbo city as an example, this study verified through a comparative experiment that the rebalancing scheme proposed can improve the operational efficiency of the bike-sharing system and reduce the management cost. Such an undertaking is performed to support the related departments in formulating reasonable rebalancing schemes for bike-sharing system.

2. Literature Review

2.1. Rebalancing Schemes. In recent years, researches related to bike-sharing system have been advancing, and these studies have provided a large number of ideas for rebalancing bike-sharing system. The main options to rebalance bike-sharing system include user-based and operator-based strategy [15]. The former is more popular in dockless bike-sharing system, while the latter performs better in station-based bike-sharing system. For the latter, current trend shows a hybrid solution in combination with different rebalancing modes.

Yin et al. [16] developed a logit model to estimate the influence of sociodemographic, land-use, and public transit characteristics on rebalancing bike-sharing system and got good result. But the logit model is too simple to deal with the complex pattern in rebalancing bike-sharing system. Lu [17] treated the issue of providing service to as much users as possible as a bike-dock pair allocation problem. They further developed an algorithm with local ratio to solve the problem. Results showed that the algorithm could save time for trips and help more users to use bike-sharing system. But if users are unwilling to walk a long way to the station, the algorithm would not work. Shi [18] designed an improved particle swarm optimization algorithm and formulated a corresponding model for rebalancing bike-sharing. However, the trucks generally have a low ratio of the average load according to the statistics. So, the algorithm needs further optimization. Liu and Ren [19] proposed a dynamic rebalancing strategy with an improved genetic algorithm. This algorithm aims to improve the overall efficiency by reducing the number of routes. But some idealized settings have caused a lot of errors in the model and reduced the effect. Mao [20] proposed a dynamic rebalancing model for bike-sharing system and calculated the number of bikes in each area with spatiotemporal graph to analyze the mobility patterns. Results showed that the proposed method had high accuracy. But there are still factors unconsidered in the model which will affect the accuracy. Recently, some scholars [21] proposed a bottom-up cluster-based model to rebalance bike-sharing system in combination with the spatiotemperal characteristic of the bike-sharing trips. However, this method is purely a static rebalancing strategy, and its effect and performance need to be improved.

2.2. Model of Level of Detail. The model of level of detail (LoD) originated in the field of computer vision. The computer determines the allocation of resources to render objects according to the position and importance in the environment so as to reduce the level of detail of non-important objects and obtains higher rendering efficiency. In recent years, the model of level of detail has been widely used in many fields.

Fan et al. [22] proposed a hierarchy training strategy on images in classification. The network would be more stable and robust through hierarchy training. Experiments show that the network trained by this strategy will get better result. Wang [23] presented a reliability-oriented hierarchical control strategy. This strategy is adopted to further improve the efficiency of the microgrids. Results show that the proposed hierarchical control structure has better performance towards reliability enhancement. The image pyramid model is another application of the level of detail model in the storage and management of the images [24]. The image pyramid establishes a series of layers with different resolutions and establishes a corresponding spatial indexing mechanism. This will improve the rendering speed of zooming or browsing images. Hu [25] further introduced a routing strategy with hierarchical model and calculated the optimal travel route with a hierarchical manner. Nakano [26] classified the transposable elements as a hierarchical classification issue and proposed hierarchical classification strategies with new data sets. Experimental results showed that the method proposed had a more competitive result compared to those by other strategies.

In short, the data structure of the model of level of detail is easy and clear. It has been applied in many academic fields. Meanwhile, the implementation of the model is simple, so it is more suitable for rebalancing bike-sharing system in urban areas.

2.3. Modeling the Rebalance of Bike-Sharing Systems

2.3.1. Overview. To study rebalancing bike-sharing system is to determine the stations that need to be rebalanced and the number of bikes that must be transferred in or out of each station, as well as to compute the optimal path for rebalancing trucks [27].

This study assumes that the total numbers of bikes and the stakes in each station are limited to simplify the rebalancing problem. A bike is either locked at a station or being used by a user. Three states occur in each station, namely, no bikes available, no idle piles, and available bikes and idle piles. We assumed that a certain number of rebalancing centers are set up. These centers are in charge of rebalancing bikes for the stations around them. Each center is equipped with several trucks. The carrying capacity of these trucks remains constant. The rebalancing system will always

monitor the status of the bike-sharing system. When the locked ratio of the piles at the station is lower than or exceeds a certain range, the system would automatically alert the staff at the management center to rebalance the bikes.

2.3.2. Model Construction. Therefore, how to design a reasonable and efficient rebalancing scheme based on the demand from stations has been a key issue. With this scheme, trucks can pass through stations in an orderly manner and finally return to the center. In this situation, the bike-sharing system could meet the needs from users as much as possible and improve the service of bike-sharing system [28, 29]. Moreover, the cost of rebalancing bike-sharing system must be taken into account, so that the running distance by the trucks is the shortest or the total time spent is the least. This is a problem that urgently needs to be solved in the current bike-sharing system, which is also the problem that this study tends to solve.

Therefore, the rebalancing scheme proposed in this work takes the transportation cost and satisfaction from bike-sharing system as the objective functions.

(1) Transportation Cost. The truck travel must be short to minimize transportation costs. The distance covered by a truck for performing rebalance for a trip is shown in the following formula:

$$Z_1 = \sum_{\substack{i=0 \\ i \neq j}}^{n} \sum_{j=0}^{n} X_{ij} d_{ij} \omega, \tag{1}$$

where X_{ij} is a decision variable. $X_{ij} = 1$ means that the truck moves from station i to station j in a certain trip, while $X_{ij} = 0$ means that the truck does not travel from station i to station j. d_{ij} is the distance from station i to station j, and ω is the transportation cost per kilometer. N refers to the total number of the stations that need to be rebalanced in a certain trip.

(2) Satisfaction. Satisfaction can be translated into penalty costs that do not satisfy time windows. Specifically, the dispatching truck should complete the dispatch task within the time expected as soon as possible to ensure that users can avail of the services at the station. We use soft-time window to constrain the arrival time of the trucks. The penalty cost is high when the arrival time before/after the expected time is early/late. The penalty cost of the time is shown in the following formula:

$$Z_2 = \sum_{i=1}^{n} W_i \left(epu \cdot \max\left(L_i - t_i, 0\right) + lpu \cdot \max\left(t_i - U_i, 0\right)\right), \tag{2}$$

where W_i is a decision variable indicating whether station i needs to be rebalanced. t_i is the time when the truck arrives at station i; $[L_i, U_i]$ is the time window of site i, and L_i represents the earliest time expected by site i; $U_i$is the latest time expected by site i; epu is the penalty cost for early arrival when the time window is unsatisfied; and lpu is the penalty cost for late arrival. Figure 1 shows the diagram of the penalty function for the soft-time window.

In summary, the objective function for the rebalancing model is shown as follows:

$$T = \alpha Z_1 + \beta Z_2, \tag{3}$$

where α is the weight value of the transportation cost, β is the weight value of penalty cost of time window, Z_1 is the transportation cost, and Z_2 is the penalty cost of the time window. Thus, the objective function to be optimized is shown as follows:

$$\alpha \left(\sum_{\substack{i=0 \\ i \neq j}}^{n} \sum_{j=0}^{n} X_{ij} d_{ij} \omega \right) + \beta \left(\sum_{i=1}^{n} W_i \times \left(epu \cdot \max\left(L_i - t_i, 0\right) + lpu \cdot \max\left(t_i - U_i, 0\right)\right) \right) \longrightarrow \text{Min.} \tag{4}$$

The constraints of the above formula are as follows:

(1) Decision variable is a 0-1 variable, which is defined as follows:

$$X_{ij} = \begin{cases} 1 & \text{if the rebalancing truck runs from station } i \text{ to } j \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

(2) In each trip, every station can only be served once by a truck. Thus,

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} X_{ij} = 1, \quad \forall j \in V \backslash \{0\}, \tag{6}$$

$$\sum_{\substack{j=0 \\ i \neq j}}^{n} X_{ij} = 1, \quad \forall i \in V \backslash \{0\}, \tag{7}$$

where V represents the collection of all the stations, including the rebalancing center which is

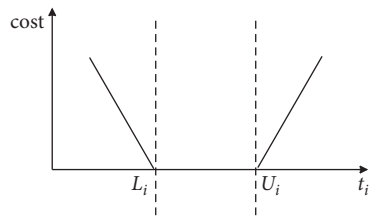


Figure 1: Penalty cost for the soft-time window.

represented by {0}. The rebalancing center can be visited multiple times in each trip.

(3) The truck would visit bike stations one by one.

$$\sum_{\substack{i=0 \\ i \neq p}}^{n} X_{ip} = \sum_{\substack{j=0 \\ j \neq p}}^{n} X_{pj}, \quad \forall p \in V\{0\}. \tag{8}$$

(4) The loading of each truck must be a nonnegative integer and does not exceed its maximum capacity.

$$\begin{gathered} g_{ij} \in Z^{+}, \\ g_{ij} \leq Q \times X_{ij} \, \forall i, j \in V, \, i \neq j, \end{gathered} \tag{9}$$

where g_{ij} represents the number of bikes that the truck transports from station i to station j, and Q is the maximum number of bikes that the truck can carry.

(5) The number of bikes is balanced before and after the station is rebalanced.

$$\sum_{\substack{j=0 \\ i \neq j}}^{n} g_{ji} + p_i = \sum_{\substack{j=0 \\ i \neq j}}^{n} g_{ij} + q_i, \quad \forall i \in V\backslash\{0\}, \tag{10}$$

where p_i is the initial number of the bikes before the station is rebalanced, and q_i is the number afterward.

(6) The arrival time t_j of the truck at station j is shown in the following formula:

$$t_j = \sum_{i=0}^{n} X_{ij}\left(t_i + tt_{ij} + \left|p_i - q_i\right| \times wt\right), \quad j \in V\backslash\{0\}, \tag{11}$$

where wt is the time of loading or unloading a bike, and tt_{ij} is the running time of the truck from station i to station j.

(7) The truck shall visit the station one by one, as shown in formula (12), where M is a number that approaches infinity.

$$t_i + tt_{ij} + \left(1 - X_{ij}\right) \times M \leq t_j, \quad \forall i \in V, \, j \in V. \tag{12}$$

In summary, the model of the rebalancing algorithm for bike-sharing system proposed in this study is shown as follows:

$$\alpha\left(\sum_{i=0}^{n} \sum_{\substack{j=0 \\ i \neq j}}^{n} X_{ij} d_{ij} \omega\right) + \beta\left(\sum_{i=1}^{n} W_i \times \left(epu \cdot \max\left(L_i - t_i, 0\right) + lpu \cdot \max\left(t_i - U_i, 0\right)\right)\right) \longrightarrow \text{Min},$$

$$\text{st.} \begin{cases} \sum_{\substack{i=0 \\ i \neq j}}^{n} X_{ij} = 1 \quad \forall j \in V\backslash\{0\} \\ \sum_{\substack{j=0 \\ i \neq j}}^{n} X_{ij} = 1 \quad \forall i \in V\backslash\{0\} \\ \sum_{\substack{i=0 \\ i \neq p}}^{n} X_{ip} = \sum_{\substack{j=0 \\ j \neq p}}^{n} X_{pj} \quad \forall p \in V\backslash\{0\} \\ g_{ij} \in Z^{+} \text{ and } g_{ij} \leq Q \times X_{ij} \quad \forall i, j \in V, i \neq j \\ \sum_{\substack{j=0 \\ i \neq j}}^{n} g_{ji} + p_i = \sum_{\substack{j=0 \\ i \neq j}}^{n} g_{ij} + q_i \forall i \in V\backslash\{0\} \\ t_j = \sum_{i=0}^{n} X_{ij}\left(t_i + tt_{ij} + \left|p_i - q_i\right| \times wt\right) \quad j \in V\backslash\{0\} \\ t_i + tt_{ij} + \left(1 - X_{ij}\right) \times M \leq t_j \quad \forall i \in V, j \in V \\ X_{ij} = \{0, 1\} \end{cases}, \tag{13}$$

2.4. K-Means. Clustering is an unsupervised learning method in machine learning. This method uses the similarity between objects to divide the entire data set into several different clusters. Such step is conducted to ensure that the data in the same cluster has a great degree of similarity, and the similarity among data in different clusters is low. In this manner, clustering is often used to reveal the inherent nature among data. The topology relationship of stations is an important spatial feature because the distribution of bike stations is regarded as a network topological structure. A reasonable division scheme is easy to design when dividing the areas where stations locate if the characteristic of the network topology is considered. In this study, we use the K-means clustering algorithm to cluster the stations on the basis of the spatial characteristics of the stations.

K-means algorithm is a commonly used clustering algorithm and an important analytical tool in research fields. This algorithm aims to randomly select k points in the data set as the initial cluster centers and then calculate the distance between each center and the remaining objects. The remaining data are then divided into the clusters closest to these k centers. Finally, the center of each cluster is recalculated according to the objects in the cluster, and the cluster is redivided. This process is repeated until the result of each division remains the same or the sum of the squared errors of the points in each cluster reaches the minimum, and the algorithm ends.

In K-means, the selection of the k value directly affects the final clustering results. Many strategies in the literature can be adopted to determine the k value. In this work, we will use the elbow rule. The core index of the elbow rule is the sum of squared errors (SSE). This index indicates the quality of the clustering result. The degree of aggregation of the clusters would increase and the value of SSE would naturally become smaller as the value of k increases [30]. Specifically, if the value of k is smaller than the real number of the clusters, the decline in SSE would be large since the increase of k will increase the aggregation of each cluster. However, if k becomes closer to the real number of the clusters, the reward in increasement of the aggregation will decrease sharply [30]. Accordingly, the decline of SSE will be sharply reduced. As the value of k continues to increase, it tends to become flattened. This indicates that the relationship between SSE and k looks like the shape of an elbow. The value of k that corresponds to the elbow part is the best clustering number, as shown in Figure 2.

The curve in Figure 2 is similar to an elbow, and the k value corresponding to the elbow is the appropriate value.

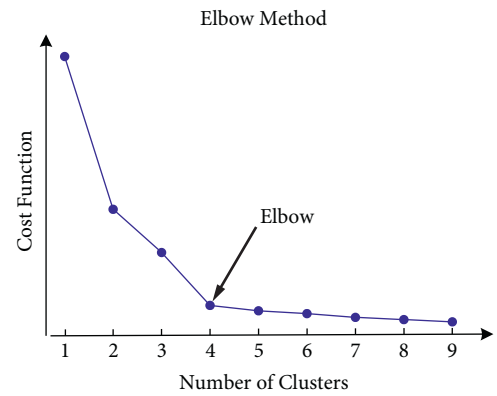


Figure 2: The elbow method.

2.5. Genetic Algorithm. Genetic algorithm (GA) is used to "search for the optimal solution by simulating the natural evolution process" [31, 32]. Some of the core operations in the genetic algorithm will be briefly introduced in this part.

2.5.1. Coding. A sequence of numbers are used for encoding to overcome such shortcomings as large search space and long codes generated by binary encoding. In this article, the number 0 is used to represent the rebalancing center and $\{1,2,3, \ldots, n\}$ to represent the bike stations that need to be rebalanced. For example, if a chromosome is [0, 2, 3, 4, 0, 1, 5, 0], it means that the rebalancing truck departs from the center, rebalances stations 2, 3, and 4 in turn, and returns to the center. Then it departs from the center to rebalance stations 1 and 5 and returns to the center again. The specific route is as follows:

Rebalancing center $\longrightarrow$ station 2 $\longrightarrow$ station 3 $\longrightarrow$ station 4 $\longrightarrow$ rebalancing center $\longrightarrow$ station 1 $\longrightarrow$ station 5 $\longrightarrow$ rebalancing center

2.5.2. Initial Population. First, the numbers of all the stations that need to be rebalanced are randomly sorted, and then $k-1$ 0 s are randomly inserted into the chromosome, where k is the number of rebalancing trucks. Then a 0 is inserted, respectively, at the beginning and at the end of the sequence, representing that the rebalancing trucks depart from the center and finally return to the center. After that, it is judged whether the current rebalancing scheme meets the capacity constraint of the truck and whether there are continuous zeros in the current rebalancing scheme. If the load constraint meets the requirement and there are no consecutive 0s in the rebalancing path, it will be chosen as an initial chromosome or it will be discarded and a new chromosome is regenerated. The above process is repeated to generate N chromosomes to form the initial population.

2.5.3. Fitness Function. Fitness can be used to simply evaluate the quality of a chromosome. The greater the value of the fitness function is, the better the corresponding chromosome is. Therefore, in this study, the reciprocal of the value of the cost function above is taken as the fitness function; namely,

$$\text{fit}(i) = \frac{1}{Z}. \tag{14}$$

2.5.4. Selection. The elite retention and roulette wheel selection are combined in this paper, which not only ensure that the best individual survives into the next

generation but also enable individuals with large value of the fitness function to enter the next generation. It is assumed that the number of chromosomes in the current population is NP, and the specific operation steps are as follows:

(1) The optimal individual with the greatest value of fitness function in the current population is recorded as *elite*. This individual does not perform pairwise crossover and mutation but is directly copied to the next generation.

(2) The roulette wheel selection is performed on the remaining individuals. In other words, the NP−1 chromosomes are selected for crossover.

2.5.5. Crossover. The crossover operation is performed on the NP−1 chromosomes selected in the previous step. New individuals are continuously generated through crossover, expanding the search space of the algorithm .

(1) When the number of the rebalancing trucks $K > 1$, this study adopts an improved crossover operator. Specific steps are shown in Figure 3.

Step 1: gene fragments with 0 at the beginning and at the end are randomly selected on the chromosomes of parent 1 and parent 2, respectively.

Step 2: prepend the gene other than the last 0 in the two gene fragments to the head of the parent chromosome.

Step 3: based on gene fragment 1 and gene fragment 2, offspring 1 and offspring 2 are constructed.

Step 4: expand offspring 1: traverse the genes of parent 2 from the left to the right, and add the genes that do not appear in gene segment 1 (offspring 1) to offspring 1 in order, and finally add code 0 at the end of offspring 1. Fill in offspring 2 symmetrically in the same way.

Step 5: for offspring 1, $K-2$ zeros are randomly inserted after segment 1 to ensure that there are K rebalancing paths. After that, it is judged whether the current chromosome meets the capacity constraint of the rebalancing truck and ensured that there are no consecutive 0s. If the constraints are not met, the $K-2$ zeros are reinserted. Offspring 2 is processed in the same way.

(2) When the number of the rebalancing trucks $K = 1$, the cyclic crossover operator is used, and the specific steps are shown in Figure 4.

Step 1: first, the 0s at the beginning and at the end of parent 1 and parent 2 are deleted, and then a position is randomly selected on parent 1.

Step 2: find the number of the genes on the corresponding position in parent 2; and then return to parent 1 to find the gene position with the same number. Repeat this process until a ring is formed. The gene and the corresponding position of parent 1 in the ring are saved.

Step 3: the gene selected in parent 1 is picked to generate offspring 1, and the position order is not disturbed.

Step 4: the remaining genes in parent 2 are put into offspring 1.

Finally, add 0 to both ends of the chromosome to get the final result of chromosome of offspring 1. Similarly, chromosome of offspring 2 can be obtained.

2.5.6. Mutation. In this study, swapping mutation is selected to perform mutation on chromosomes; namely, two nonzero genes are randomly selected in the parent chromosomes and exchanged. If the route could meet the constraint of truck capacity after the exchange, it will updated.

In summary, the complete algorithm flow of genetic algorithm is shown in Figure 5.

3. Rebalancing Strategy Based on the Model of Level of Detail

3.1. K-Divisive Hierarchical Clustering Algorithm. The divisive hierarchical clustering algorithm [33, 34] is also called the top-down clustering method. In this algorithm, all objects in the collection are treated as a cluster first. This cluster is then divided into small ones, each as a new cluster to be subdivided. Overall, the implementation of the divisive hierarchical clustering method is also a process of constructing a tree from top to bottom, as shown in Figure 6, where lines with different colors refer to different clusters. In the aforementioned recursive division process, all objects are first treated as the root node of a tree. The cluster is then recursively split into multiple small ones. This process is iteratively executed until only one object is present in each cluster or a predefined termination condition is satisfied.

Among the many divisive hierarchical clustering algorithms, the divisive hierarchical clustering with K-means, which is called the K-divisive hierarchical clustering, is one of the most widely used algorithms. This algorithm aims to divide data by using the divisive hierarchical clustering algorithms through iteratively applying K-means until each object becomes a cluster or the termination conditions are satisfied.

The K-divisive hierarchical clustering method combines the traditional K-means algorithm with divisive hierarchical clustering, thereby not only making the clustering realistic but also greatly improving the clustering efficiency [35].

3.2. Rebalancing Algorithm Based on the K-Divisive Hierarchical Clustering. The corresponding rebalancing strategy for bike-sharing system is designed in this section on the basis of the K-divisive hierarchical clustering algorithm. The previous analysis indicates that the K-divisive hierarchical clustering algorithm is actually a process of dividing the cluster with granularity from coarse to fine. The essence of the rebalancing algorithm proposed is a rebalancing strategy with granularity from large to small and a continuous refinement process.

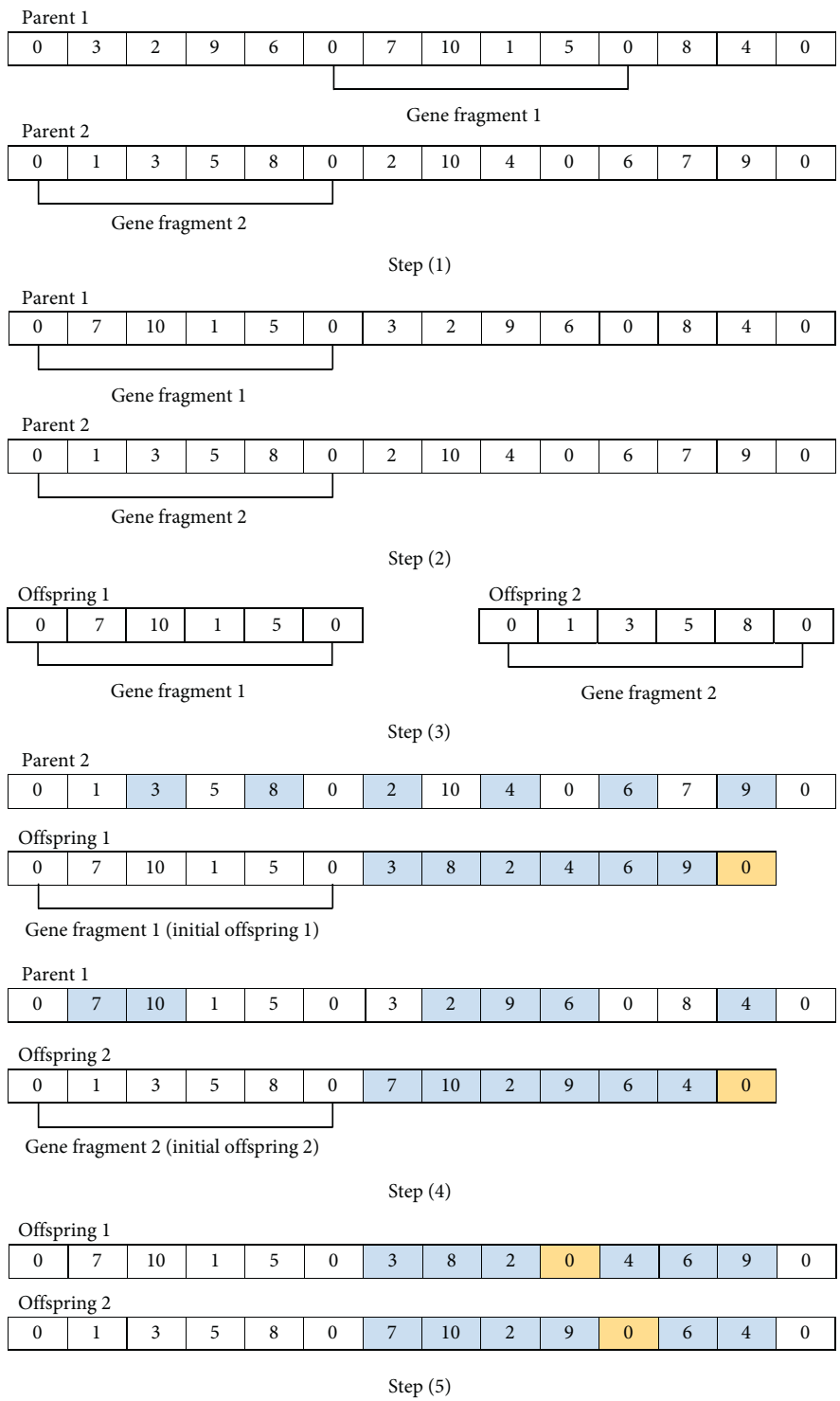


Figure 3: The crossover operation when $k > 1$.

After the corresponding divisive areas are obtained, the areas covered by clusters, which are divided from the original data set, are treated as rebalancing units on the top level. The renting/returning number of bikes in each cluster is then counted. Based on the statistic result, a corresponding rebalancing scheme is formulated, which is also the rebalancing strategy on the top level. Then, the subclusters of the bike stations on the next level are obtained for each rebalancing unit. The clusters at this level are more refined than the previous ones. The corresponding rebalancing units are formed according to the range of each cluster. These units are treated as rebalancing cells on the second level. Thereafter, all clusters on different levels are traversed according to such a rule. The corresponding rebalancing scheme is

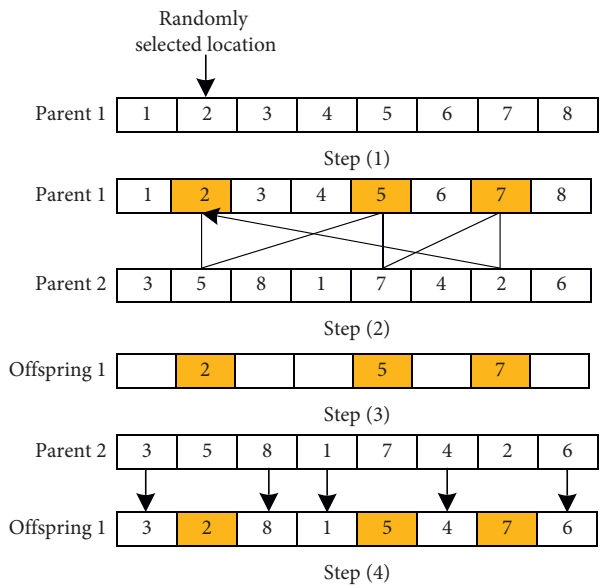


FIGURE 4: The crossover operation when $k = 1$.

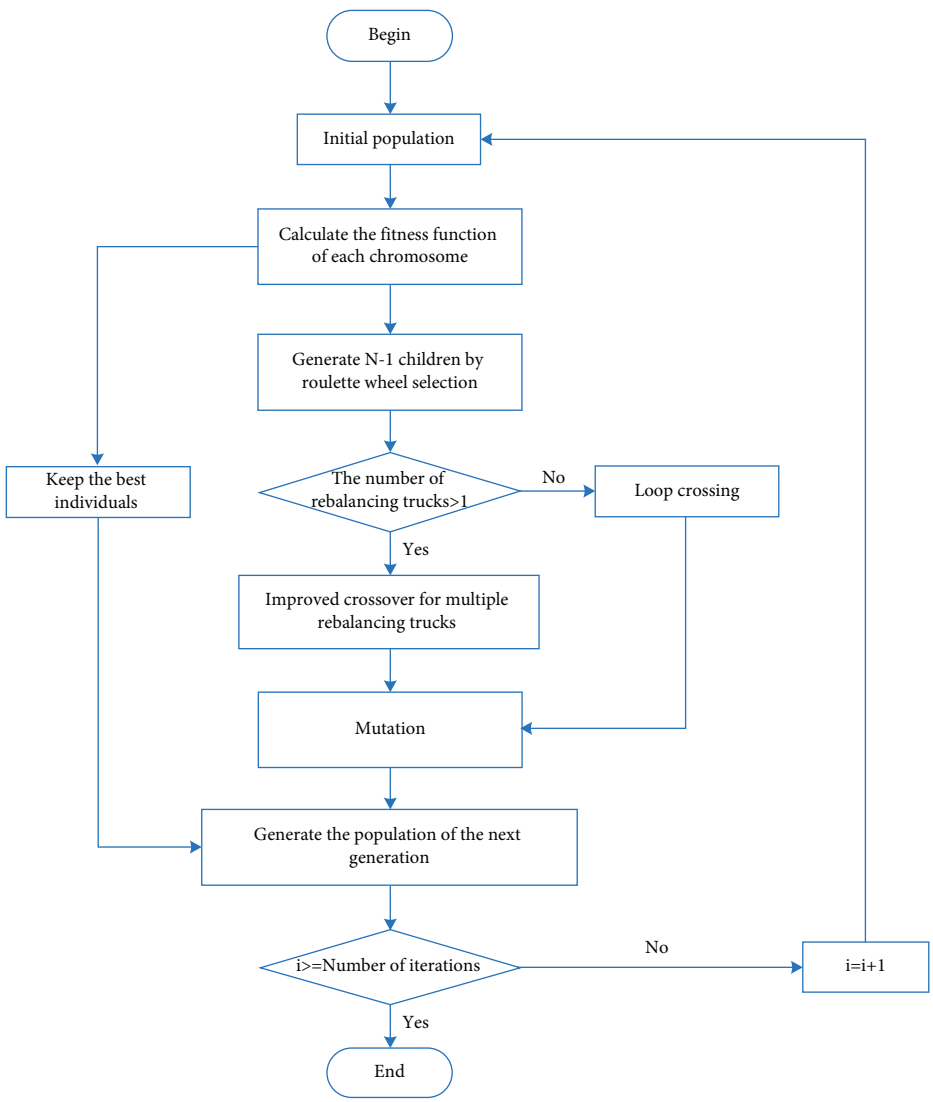


FIGURE 5: Flow of the genetic algorithm.

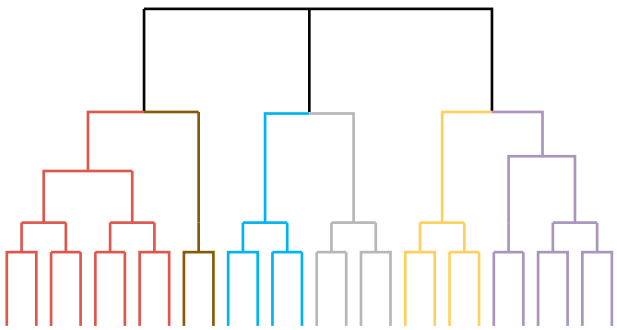


Figure 6: K-divisive hierarchical clustering algorithm.

computed according to the renting/returning demand among different clusters on the same level. In this way, the rebalancing strategies on each layer are integrated together to form a tree-like scheme, which is the rebalancing algorithm based on the model of level of detail. The complete process is shown in Figure 7.

In summary, the proposed algorithm is actually a rebalancing process from coarse to fine. This notion also fully reflects the cognitive process of humans from the macro to the micro. When decision-makers are conducting rebalancing schemes, they always carry out the scheme from a macro perspective and then gradually refine it to small ones, rather than focusing on the needs of one or two stations from the beginning. Therefore, the rebalancing algorithm proposed in this work also conforms to the processes and characteristics of cognition and thinking.

4. Results and Discussion

The experiments were conducted using Python, and the program was tested and run under the operating system of Windows 10. Besides, a comparison experiment was performed with the traditional scheme. The results showed that the rebalancing algorithm proposed in this work had obvious advantages compared with the traditional one.

4.1. Experiment for K-Divisive Hierarchical Clustering. The bike-sharing system in the downtown area of Ningbo City is selected as the experimental area in this work. Statistics shows that 169 bike stations are positioned in the area. The area where these stations located is divided into multiple levels of detail, as shown in Figure 8, where one dot represents a bike station. Specifically, the collection of all the stations was regarded as a cluster, which was also treated as the start of the hierarchical division. These stations were used to draw an elbow diagram. A suitable k value was found to divide the cluster that composed of all stations to form small subclusters. These subclusters were regarded as cells on the top level, as areas 1, 2, and 3 shown in Figure 8.

The entire area where all the bike stations locate is divided into three subclusters. These clusters generally cover a large space. The number of bike stations included is relatively large. These stations have a low-density distribution, and the arrangement between stations is relatively sparse. Thereafter, the elbow diagram is drawn again to find the appropriate k value in each subcluster. The K-means clustering algorithm is applied to further divide the subcluster to obtain a second-level division scheme (e.g., areas 11, 12, and 13 in Figure 8).

The subcluster range will gradually diminish with continuous division. The number of bike stations in each subcluster will also decrease. The division process is continuously performed according to such rules until each cluster contains a small number or only one station. A hierarchical division scheme for stations in bike-sharing system with granularity ranging from large to small is finally formed by overlaying together multiple layers of clusters with different extensions.

4.2. Experiment for Rebalancing Algorithm Based on the Model of LoD. To minimize the amount of calculation, the clusters formed by the stations on the bottom in the division scheme of LoD are treated as the rebalancing units on the last level. In other words, the stations in these cells are no longer to be divided. Combined with the renting/returning demand in each cell, the rebalancing scheme on each layer is computed. Then the rebalancing schemes at each layer are superimposed together to form a complete scheme with granularity from coarse to fine.

The records in rush hours in the morning on June 30, 2019, were selected as manifested in the sample data in the study experiment. The number of bikes at each station at 7:00 a.m. was treated as the initial state. The usage of the bikes at each station was simulated by reading the rental/returning records in the database. The alarm thresholds at stations were set to [0.4, 0.6]. Specifically, the station would alarm when its saturation exceeds the threshold interval. Thereafter, a fixed time window starting from the alarming time was generated. The total demand for bikes in each area and its corresponding time window were determined separately, as shown in Table 1.

The genetic algorithm was applied to compute the optimal path among rebalancing units. The times that trucks return to the rebalancing center should be minimized as much as possible. After the rebalancing task at the current level was completed, rebalance for units on the lower level was further performed. Our genetic algorithm supposes the following: the population size is 100, the number of iterations is 1000, the mutation rate is 0.1, the running speed of the truck is 30 km/h, and the maximum carrying capacity of the truck is 400 with an initial load of 250. Finally, the rebalancing schemes at different levels were solved, as shown in Table 2.

4.3. Comparative Experiment and Analysis. We also compared the proposed algorithm with the traditional one. In the traditional algorithm, we did not divide the area where the station locates with the model of LoD. Instead, the genetic algorithm is directly called to rebalance the areas on the lowest layer in the model of LoD of the clusters.

In this experiment, we only assigned one truck to rebalance the bike-sharing system to clarify the results. It is supposed that the running speed of the truck is 30 km/h. The

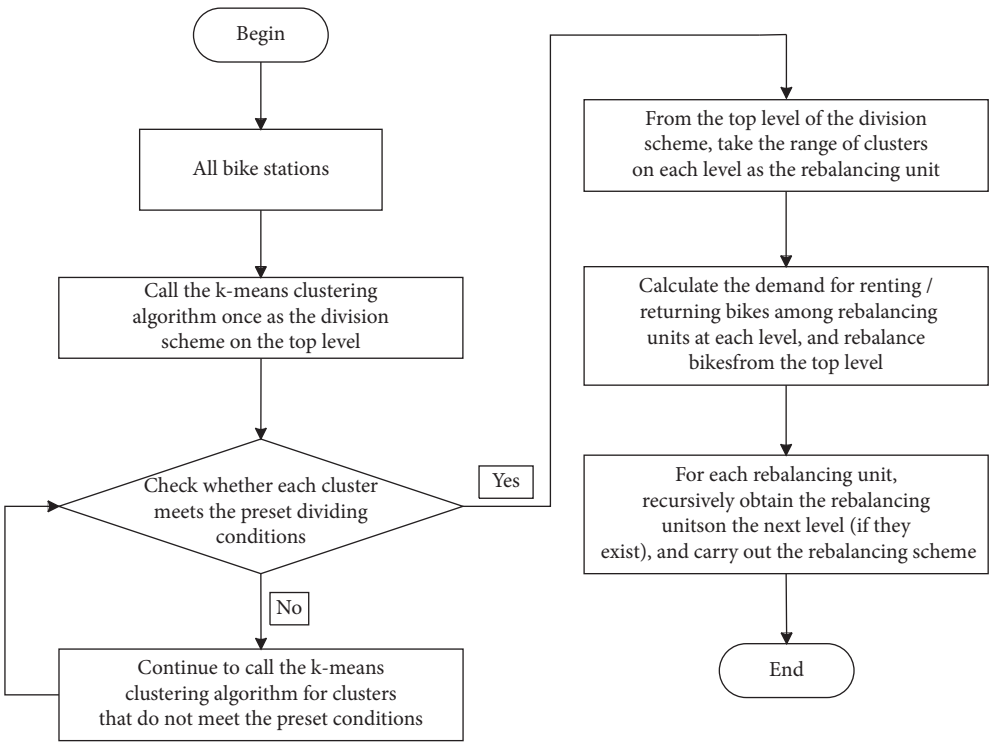


Figure 7: Flowchart of the proposed algorithm.

Figure 8: Hierarchical division scheme for bike stations in the downtown area of Ningbo.

Table 1: Demands for bikes and the corresponding time windows in each rebalancing unit.

Area number	Number of piles	Demand	Time window
C1	1702	170	7:46--8:46
C2	1696	−134	8:16--9:16
C3	1341	−70	8:20--9:20
C11	697	−35	8:15--9:15
C12	340	67	7:53--8:53
C13	665	38	7:43--8:43
C21	383	41	7:45--8:45
C22	405	46	7:43--8:43
C23	450	46	8:07--9:07
C24	458	−29	7:48--8:48
C31	280	39	8:05--9:05
C32	391	−39	8:24--9:24
C33	385	29	8:05--9:05
C34	285	170	7:46--8:46
C111	125	−13	7:52--8:52
C112	120	13	7:41--8:41
C113	192	20	7:50--8:51
C115	115	−12	7:50--8:50
C121	85	−9	8:01--9:01
C123	130	14	8:18--9:18
C132	105	11	7:51--8:51
C133	85	9	7:46--8:46
C134	155	16	7:43--8:43
C135	130	14	7:52--8:52
C211	75	8	7:48--8:48
C212	113	11	7:42--8:42
C213	195	20	7:42--8:42
C221	150	16	7:44--8:44
C222	130	−14	7:55--8:55
C223	125	13	7:47--8:47
C231	195	20	7:37--8:37
C242	70	8	7:37--8:37
C244	106	11	7:45--8:45
C311	160	−17	7:51--8:51
C312	120	−13	7:43--8:43
C321	140	−15	7:41--8:41
C322	115	12	7:38--8:38
C323	136	−14	7:45--8:45
C331	210	−22	7:49--8:49
C332	175	18	7:51--8:51
C341	55	6	7:39--8:39
C342	55	6	7:37--8:37
C343	175	18	8:01--9:01

experimental results are shown in Figure 9. Figure 9(a) shows that the truck would directly run towards all rebalancing units from the center with the traditional rebalancing algorithm. Figure 9(b) shows the rebalancing scheme based on model of LoD proposed in this work.

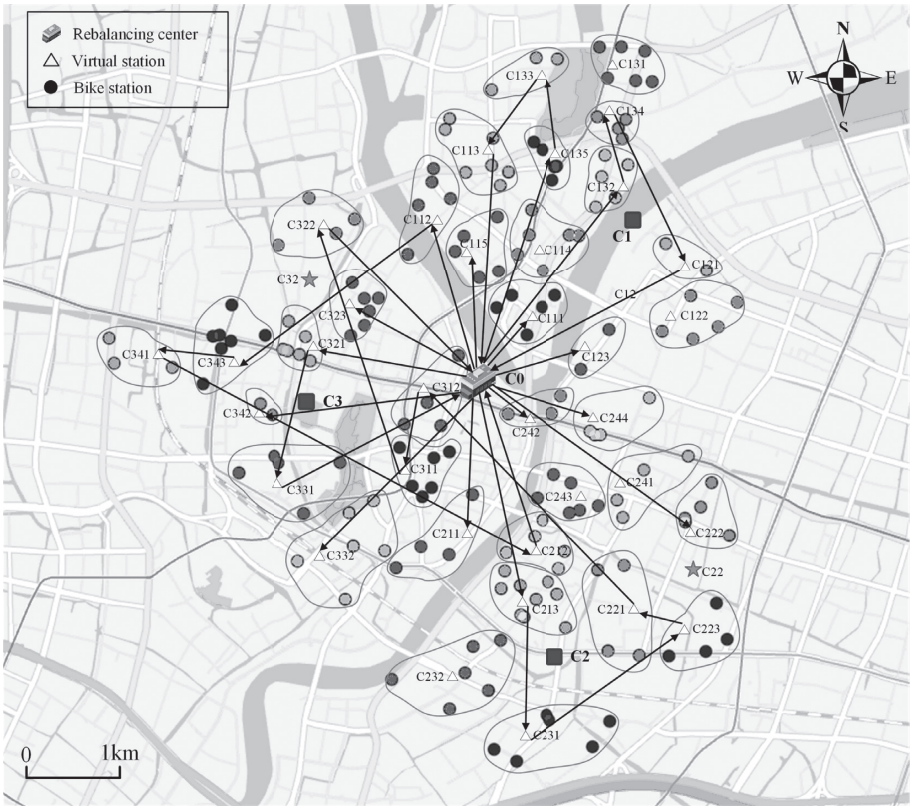
If the traditional rebalancing method is adopted, then the truck would directly run to all stations from the rebalancing center. The rebalancing scheme for each station was calculated by the genetic algorithm, as shown in Table 3. The total length covered by the truck is 65318.75 m, and the effective time is 130.7 min as shown in Table 4. However, on the basis of the rebalancing scheme proposed in this work, we collected the statistical data on different levels and found that the total length of the path using the proposed method in this paper is 46845.62 m, and the effective time is 93.73 min as shown in Table 4. It is concluded that the proposed algorithm can reduce the effective time for rebalancing bike-sharing system by 28.3% compared with the traditional one.

The rebalancing method proposed in this work can effectively shorten the total distance, thereby reducing the effective time. This mechanism can help to satisfy the needs of users to use bikes as much as possible and improve the overall service of bike-sharing system in urban areas. Therefore, this rebalancing scheme is practical for bike-sharing system.

The rebalancing algorithm is the key to solving the imbalance distribution of the demand for bikes in the space and time. The experiment proves that the rebalancing method proposed in this work can upgrade the response speed, improve the service of the bike-sharing system, and

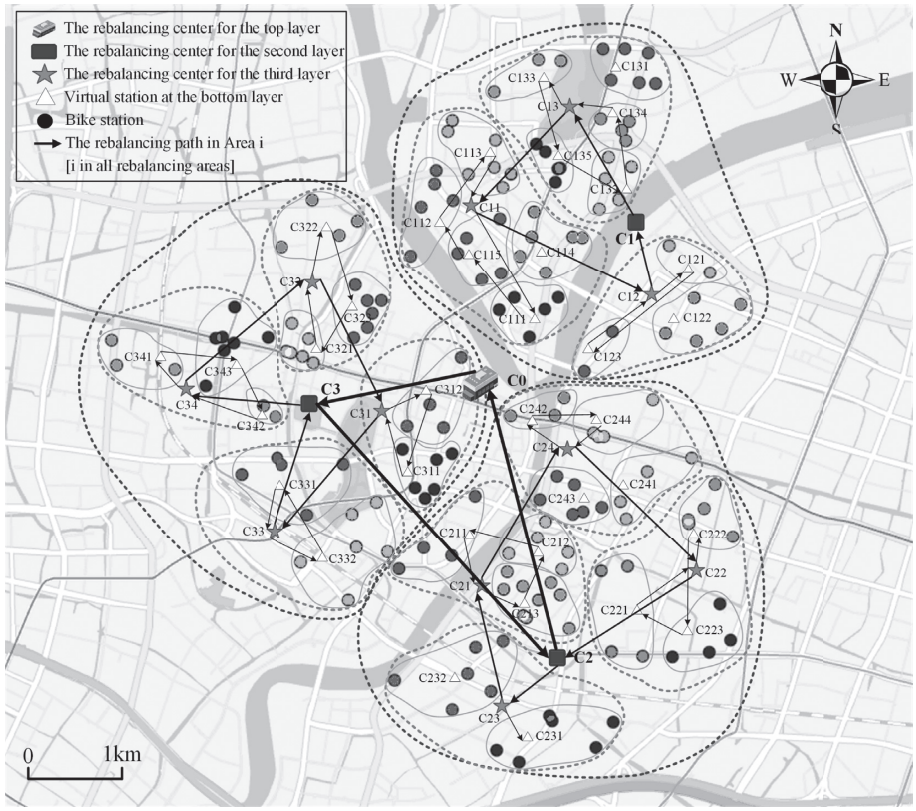
Table 2: Rebalancing schemes computed by the proposed algorithm.

Rebalancing area	Rebalancing path	Number of bikes that rebalanced	Cumulative distance/m
Original area	C0 ⟶ C3 ⟶ C2 ⟶ C0	304	5845.56
1	C1 ⟶ C13 ⟶ C11 ⟶ C12 ⟶ C1	172	4948.04
11	C11 ⟶ C111 ⟶ C115 ⟶ C112 ⟶ C113 ⟶ C11	58	3266.56
12	C12 ⟶ C121 ⟶ C123 ⟶ C12	23	2298.27
13	C13 ⟶ C133 ⟶ C135 ⟶ C132 ⟶ C134 ⟶ C13	50	2780.79
2	C2 ⟶ C23 ⟶ C21 ⟶ C24 ⟶ C22 ⟶ C2	171	5725.94
21	C21 ⟶ C213 ⟶ C212 ⟶ C211 ⟶ C21	39	2402.29
22	C22 ⟶ C222 ⟶ C223 ⟶ C221 ⟶ C22	43	3086.06
23	C23 ⟶ C231 ⟶ C23	20	952.48
24	C24 ⟶ C242 ⟶ C244 ⟶ C24	19	1505.56
3	C3 ⟶ C34 ⟶ C32 ⟶ C31 ⟶ C33 ⟶ C3	136	5740.13
31	C31 ⟶ C312 ⟶ C311 ⟶ C31	30	1201.95
32	C32 ⟶ C322 ⟶ C323 ⟶ C321 ⟶ C32	41	2629.86
33	C33 ⟶ C332 ⟶ C331 ⟶ C33	40	1660.51
34	C34 ⟶ C341 ⟶ C343 ⟶ C342 ⟶ C34	30	2801.62



(a)

Figure 9: Continued.



(b)

Figure 9: Comparison of the traditional rebalancing scheme and the one proposed in this work. (a) Traditional rebalancing scheme. (b) Rebalancing scheme proposed in this work.

Table 3: Length of the rebalancing path and time by the traditional method.

Path	Distance (m)	Time consumed (min)
Path 1	5845.10	11.69
Path 2	2124.06	4.25
Path 3	2379.34	4.76
Path 4	1481.96	2.97
Path 5	4789.14	9.58
Path 6	5701.86	11.41
Path 7	900.13	1.80
Path 8	3961.12	7.93
Path 9	1898.28	3.80
Path 10	3679.45	7.36
Path 11	2843.11	5.69
Path 12	2623.59	5.25
Path 13	4244.56	8.49
Path 14	12828.01	25.67
Path 15	10019.04	20.05
Total	65318.75	130.7

Table 4: Length of the rebalancing path and time by the proposed method.

Area	Distance (m)	Time consumed (min)
Original area	5845.56	11.70
1	4948.04	9.90
11	3266.56	6.54
12	2298.27	4.60
13	2780.79	5.56
2	5725.94	11.46
21	2402.29	4.81
22	3086.06	6.17
23	952.48	1.91
24	1505.56	3.01
3	5740.13	11.48
31	1201.95	2.40
32	2629.86	5.26
33	1660.51	3.32
34	2801.62	5.61
Total	46845.62	93.73

reduce the maintenance cost. If the method proposed in this work is applied to the operation and management of the bike-sharing system in major cities, then it will not only help improve the service of such system and increase the satisfaction from citizens but also effectively guide people to use bike-sharing system instead of cars. Accordingly, the current situation of traffic congestion on the road is enhanced.

5. Conclusion

We studied a rebalancing scheme for bike-sharing system on the basis of the model of level of detail combined with the K-divisive hierarchy clustering algorithm. The algorithm takes the comprehensive consideration of maximizing to satisfy the renting or returning demand for bikes and minimizing the rebalancing cost as the goals to be optimized.

The algorithm initially treats the collection of all stations as a cluster and then uses the K-means algorithm to cluster them. The obtained subclusters are then reclustered using K-means to form small subclusters. Clustering is continuously executed according to such a rule to form hierarchical layers. The areas where bike stations locate are divided with different granularities. In each layer of the hierarchical structure, the extent covered by the cluster is regarded as a rebalancing unit. Finally, genetic algorithm is called to compute the optimal rebalancing route among units at each level. A rebalancing scheme from coarse to fine is formed from a macro perspective.

Meanwhile, we take bike-sharing system in the downtown area of Ningbo as an example to verify the feasibility and practicability of the algorithm. The comparative experiments proved that the rebalancing scheme can effectively relieve the problem of "difficult to rent/return a bike" in the bike-sharing system. Compared with the traditional method, the algorithm proposed can reduce the effective time for rebalancing bike-sharing system by 28.3%. Therefore, this work can provide an effective theoretical basis for rebalancing bike-sharing system in urban areas and important guidance for management departments.

Although this work has achieved considerable results, some shortcomings are also noted. (1) In establishing the model, the length of the time window is set to a fixed value. We did not discuss the setting of this value in detail. In our future research, the characteristics of demand for renting/returning bikes at stations will be comprehensively considered to analyze the effect of setting different values of time window on the algorithm. This task is necessary to determine the optimal value of the time window. (2) The experiments involved in this study are only used to verify the algorithm's effectiveness. In the future research, we will combine the spatiotemporal distribution characteristics of renting or returning demand of bikes with a neural network to predict the renting/returning requirements. By doing so, the algorithm proposed could be more accurate.

The research contribution will bring inspiration or guidance to researchers engaged in this field. We will further study the dynamic and intelligent rebalancing strategies for the bike-sharing system in urban areas.

Data Availability

The sample data and codes that support the findings of this study are available at https://figshare.com/s/b1f7067d0e6ce05bb003.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the Natural Science Foundation of Zhejiang Province (no. LQ18D010008), Natural Science Foundation of Ningbo (no. 2018A610132), and a project supported by Scientific Research Fund of Zhejiang Provincial Education Department (Y201736984).

References

[1] Y. Su, X. Liu, and X. Li, "Research on traffic congestion based on system dynamics: the case of Chongqing, China," *Complexity*, vol. 2020, 2020.

[2] H. Luo, "Optimizing bike sharing systems from the life cycle greenhouse gas emissions perspective," *Transportation Research Part C: Emerging Technologies*, vol. 117, Article ID 102705, 2020.

[3] J. Qin, S. Lee, X. Yan, and Y. Tan, "Beyond solving the last mile problem: the substitution effects of bike-sharing on a ride-sharing platform," *Journal of Business Analytics*, vol. 1, no. 1, pp. 13–28, 2018.

[4] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors*, vol. 18, no. 2, p. 512, 2018.

[5] B. Legros, "Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station," *European Journal of Operational Research*, vol. 272, no. 2, pp. 740–753, 2019.

[6] P. Yi, F. Huang, and J. Peng, "A rebalancing strategy for the imbalance problem in bike-sharing systems," *Energies*, vol. 12, no. 13, p. 2578, 2019.

[7] J. Dötterl, "Towards dynamic rebalancing of bike sharing systems: an event-driven agents approach," in *EPIA Conference on Artificial Intelligence*, Springer, Berlin, Germany, September 2017.

[8] S. J. Patel and C. R. Patel, *An Infrastructure Review of Public Bicycle Sharing System (PBSS): Global and Indian Scenario*, Springer Singapore, Singapore, 2019.

[9] J. X. Cao, C. C. Xue, M. Y. Jian, and X. R. Yao, "Research on the station location problem for public bicycle systems under dynamic demand," *Computers & Industrial Engineering*, 127, pp. 971–980, 2019.

[10] Y. Zhu and M. Diao, "Understanding the spatiotemporal patterns of public bicycle usage: a case study of Hangzhou, China," *International Journal of Sustainable Transportation*, vol. 14, no. 3, pp. 163–176, 2020.

[11] Z. Tian, J. Zhou, W. Y. Szeto, L. Tian, and W. Zhang, "The rebalancing of bike-sharing system under flow-type task window," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 1–27, 2020.

[12] G. Xu, "A mixed rebalancing strategy in bike sharing systems," *Engineering Optimization*, vol. 54, pp. 1–18, 2021.

[13] J. Sultan, G. Ben-Haim, J.-H. Haunert, and S. Dalyot, "Extracting spatial patterns in bicycle routes from crowdsourced data," *Transactions in GIS*, vol. 21, no. 6, pp. 1321–1340, 2017.

[14] C. Yang and G. Gidófalvi, "Mining and visual exploration of closed contiguous sequential patterns in trajectories," *International Journal of Geographical Information Science*, vol. 32, no. 7, pp. 1282–1304, 2018.

[15] C. M. Vallez, M. Castro, and D. Contreras, "Challenges and opportunities in dock-based bike-sharing rebalancing: a systematic review," *Sustainability*, vol. 13, no. 4, p. 1829, 2021.

[16] A. Yin, B. Ning, and Y. Wang, "Study on bike sharing rebalancing: evidence from kunming," in *Resilience and Sustainable Transportation Systems*, pp. 280–289, American Society of Civil Engineers Reston, Reston, VA, USA, 2020.

[17] P. Lu, "Local ratio based distributed bike-dock pair allocation in public bike system," in *Proceedings of the ACM Turing Celebration Conference-China*, Chengdu, China, May 2019.

[18] L. Shi, Y. Zhang, W. Rui, and X. Yang, "Study on the bike-sharing inventory rebalancing and vehicle routing for bike-

sharing system," *Transportation research procedia*, vol. 39, pp. 624–633, 2019.

[19] Z. Liu and L. Ren, "A sharing bike scheduling optimization algorithm based on two-dimensional dynamic model and improved genetic algorithm," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, Halifax, Canada, July 2018.

[20] D. Mao, "Bike-sharing dynamic scheduling model based on spatio-temporal graph," in *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE, Bangkok, Thailand, January 2018.

[21] B. Lahoorpoor, H. Faroqi, A. Sadeghi-Niaraki, and S.-M. Choi, "Spatial cluster-based model for static rebalancing bike sharing problem," *Sustainability*, vol. 11, no. 11, p. 3205, 2019.

[22] L. Fan, C. Li, and M. Shi, "Hierarchy training strategy in image classification," in *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, IEEE, Lanzhou, China, August 2018.

[23] Y. Wang, "A hierarchical control strategy of microgrids toward reliability enhancement," in *2018 International Conference on Smart Grid (icSmartGrid)*, IEEE, Nagasaki. Japan, December 2018.

[24] H. Zhenghua, M. Lingkui, and Z. Wen, "Relational database extension oriented, self-adaptive imagery pyramid model," *Acta Geodaetica et Cartographica Sinica*, vol. 44, no. 6, p. 678, 2015.

[25] Z. Hu, T. Jia, G. Wang, J. Wang, and L. Meng, "Computing a hierarchy favored optimal route in a Voronoi-based road network with multiple levels of detail," *International Journal of Geographical Information Science*, vol. 31, no. 11, pp. 2216–2233, 2017.

[26] F. K. Nakano, "Top-down strategies for hierarchical classification of transposable elements with neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Anchorage, AK, USA, May 2017.

[27] D. Chemla, F. Meunier, and R. Wolfler Calvo, "Bike sharing systems: solving the static rebalancing problem," *Discrete Optimization*, vol. 10, no. 2, pp. 120–146, 2013.

[28] W. Li, "Understanding intra-urban human mobility through an exploratory spatiotemporal analysis of bike-sharing trajectories," *International Journal of Geographical Information Science*, vol. 61, pp. 1–24, 2020.

[29] P. Zhao, "An empirical study on the intra-urban goods movement patterns using logistics big data," *International Journal of Geographical Information Science*, vol. 42, pp. 1–28, 2018.

[30] Y. Xie, "Code similarity detection technique based on AST unsupervised clustering method," in *2020 IEEE 6th International Conference on Computer and Communications*, ICCC), Chengdu, China, December 2020.

[31] S. Sivanandam and S. Deepa, "Genetic algorithms," in *Introduction to Genetic Algorithms*, pp. 15–37, Springer, Berlin, Germany, 2008.

[32] D. B. Fogel, "Applying evolutionary programming to selected traveling salesman problems," *Cybernetics & Systems*, vol. 24, no. 1, pp. 27–36, 1993.

[33] P. K. Amalaman and C. F. Eick, *Avalanche: A Hierarchical, Divisive Clustering Algorithm*, Springer International Publishing, Cham, Switzerland, 2015.

[34] N. Lavrac, "Visual divisive hierarchical clustering using k-means," 2012.

[35] M. V. Reddy, M. Vivekananda, and R. Satish, "Divisive heirarchical clustering with K-means and agglomerative heirarchical clustering," *International Journal of Computer Science Trends and Technology*, vol. 5, no. 5, pp. 6–11, 2017.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3606895, 22 pages
https://doi.org/10.1155/2021/3606895

Research Article

An Adaptive Parallel Arithmetic Optimization Algorithm for Robot Path Planning

Ruo-Bin Wang,[1] **Wei-Feng Wang**,[1] **Lin Xu**,[2] **Jeng-Shyang Pan**,[3] **and Shu-Chuan Chu**[3,4]

[1]*School of Information Science and Technology, North China University of Technology, Beijing 100043, China*
[2]*STEM, University of South Australia, Adelaide 5095, Australia*
[3]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[4]*College of Science and Engineering, Flinders University, 1284 South Road, Tonsley, South Australia 5042, Australia*

Correspondence should be addressed to Ruo-Bin Wang; robin945@163.com and Lin Xu; xuyly032@mymail.unisa.edu.au

Received 16 July 2021; Revised 15 August 2021; Accepted 3 September 2021; Published 24 September 2021

Academic Editor: Chi-Hua Chen

Path planning is one of the hotspots in the research of automotive engineering. Aiming at the issue of robot path planning with the goal of finding a collision-free optimal motion path in an environment with barriers, this study proposes an adaptive parallel arithmetic optimization algorithm (APAOA) with a novel parallel communication strategy. Comparisons with other popular algorithms on 18 benchmark functions are committed. Experimental results show that the proposed algorithm performs better in terms of solution accuracy and convergence speed, and the proposed strategy can prevent the algorithm from falling into a local optimal solution. Finally, we apply APAOA to solve the problem of robot path planning.

1. Introduction

In recent years, automotive engineering has been an emerging area. Among it, the automotive robots have been widely employed in industry and social life and played an important role, especially during the pandemic of COVID-19. The existing literatures have explored issues on robot path planning. For example, Sarabu et al. [1] proposed the method of using dual robotic arms to collaboratively pick apples in an unstructured orchard environment. Sehestedt et al. [2] utilized a path planning algorithm based on probabilistic routes to sample Hidden Markov models through robots learning human motion patterns. Tiseni et al. [3]conducted an evaluation about measuring the energy dose delivered by a robot-based moving source of Ultraviolet type-C irradiation (UV-C) radiation at different locations in an indoor environment with genetic algorithm (GA).

In summary, robot path planning has become one of the key problems in the domain of robot automatic control. At present, there are three kinds of popular algorithms employed in path planning:

(1) Based on searching: Dijkstra algorithm [4] and A* algorithm [5]

(2) Based on probability: rapidly exploring Random Trees (RRT) [6] and rapidly exploring Random Trees* (RRT*) [7]

(3) Based on metaheuristic algorithm: particle swarm optimization (PSO) [8]

In this study, a metaheuristic algorithm is employed to optimize the robot path planning. Compared with other algorithms, a metaheuristic algorithm can achieve a stable convergence and avoid trapping into a local optimal solution, especially when facing a complex environment.

In the past few decades, metaheuristic algorithms have been regarded as an effective way to address optimization issues in various fields and have been widely used to improve the performance of real-world problems. Some popular metaheuristic algorithms are ant colony optimization algorithm (ACO) [9], ant lion optimizer (ALO) [10], moth-flame optimization algorithm (MFO) [11], multiverse optimizer (MVO) [12], sine cosine algorithm (SCA) [13], whale

optimization algorithm (WOA) [14], dragonfly algorithm (DA) [15], cat swarm optimization (CSO) [16], and so on. These algorithms have been successfully applied to different fields of medical industry [17], image processing [18], transportation [19], and the like. However, according to the No Free Lunch Theorem (NFL) [20], there is no metaheuristic algorithm suitable for handling all types of optimization problems. Therefore, researchers have continuously proposed new metaheuristic algorithms in recent years and have continuously improved them to deal with increasing complexities in the real world.

Currently, there are existing researches using metaheuristic algorithm for path planning. Zhang et al. [21] proposed a new improved artificial fish swarm algorithm (IAFSA) to process the mobile robot path planning problem in a real environment. Qin et al. [22] proposed an improved PSO with mutation operator to get the optimal path. Li et al. [23] proposed an improved GA, which integrated a fuzzy logic control algorithm to self-adaptively adjust the probabilities of crossover and mutation in GA. Wang et al. [24] presented a new weighted adjacency matrix to determine the walking direction, and the best ant and the worst ant are introduced for the adjustment of pheromone to facilitate the searching process. The proposed algorithm guarantees that robots are able to find an optimal path. Zhang et al. [25] proposed hybrid method with simulated annealing algorithm and ant colony algorithm to apply to robot path planning. Huang and Tsai [26] combined GA with PSO in evolving new solutions by applying crossover and mutation operators on solutions constructed by particles, which avoids the premature convergence and time complexity. Aiming at the problems of slow response speed, long planning path, unsafe factors, and a large number of turns in the traditional path planning algorithm, Dao et al. [27] proposed a novel multiobjective method for optimal mobile robot path planning based on WOA.

Usually, the original algorithm is improved and then applied to path planning. The types of algorithm improvement are roughly divided into three categories: improvements on parameters, hybrid algorithm, and multiobjective algorithms.

The arithmetic optimization algorithm (AOA) [28] is proposed in 2021, with the characteristics of simplicity, fewer control parameters, and stronger output performance. It has been proved that AOA performs well on welded beam design problem, tension/compression spring design problem, pressure vessel design problem, and the like. However, there is still seldom research on the improvement of AOA and its application on robot path planning. The algorithm is relatively new; therefore, there are relatively few improvements on it [29, 30], which deserves further improvement in certain areas. Parallel strategy is an effective algorithm optimization method, which can communicate and exchange information among groups. Some parallel algorithms are parallel SCA [31], parallel GA [32], parallel MVO [18], parallel WOA [33], parallel SCO [34], and parallel GWO [35]. However, there is still a lack of improvements on parallel strategies for AOA. Although AOA has superiority in some aspects over some other algorithms, there are still defects of converging slowly in complex environments or high-dimensional problems and is prone to fall into local optimal. Aiming at the algorithm defects, we propose an adaptive parallel AOA. Adaptive parameters can balance the capabilities of exploration and exploitation. Parallel strategy refers to strengthening the communication among groups and reducing the defects of the original AOA, such as premature convergence, search stagnation, and easy to fall into the local optimal search space. The main contributions of this article are summarized as follows:

(1) We propose a novel parameter adaptive equation to control the AOA sensitive parameter α, which can balance the capabilities of exploration and exploitation

(2) We propose a novel parallel communication strategy and apply it to AOA, which can strengthen the communication and information exchange among groups and avoid falling into the local optimal solution

(3) The improved AOA is applied to an optimization problem of 2D robot path planning

The rest of the article is organized as follows. Section 2 describes the principle of the original AOA and robot path planning. Section 3 introduces the improved AOA about self-adaption and parallel strategy. The performance of the proposed algorithm is tested, and the results of different algorithms are shown and analysed in Section 4. Section 5 introduces the application of the proposed algorithm in robot path planning. Finally, conclusions are delivered in Section 6.

2. Related Works

2.1. Original AOA. The AOA is inspired by the application of arithmetic operators in solving arithmetic problems [28], using simple arithmetic operators, such as addition, subtraction, multiplication, and division as mathematical optimization, to search for the optimal solution that meets the standards from a set of candidate solutions. Like MVO [12], AOA is also divided into the phrases of exploration and exploitation. Exploration refers to finding a range of promising optimal solutions in a broad search space, and exploitation refers to quickly finding the optimal solution within the range of promising solutions and converging.

Before running, the search phase of AOA should be confirmed. Therefore, a coefficient calculated by equation (1) is employed for choosing the phrase of exploration or exploitation, which is Math Optimizer Accelerated (MOA) function. Another coefficient defined by equation (2) is Math Optimizer Probability (MOP), which is employed for controlling the range of candidate solutions in the phase of exploring or exploiting.

$$\mathrm{MOA}(t) = \mathrm{Min} + t \times \left(\frac{\mathrm{Max} - \mathrm{Min}}{T}\right), \tag{1}$$

$$\mathrm{MOP}(t) = 1 - \left(\frac{t}{T}\right)^{1/\alpha}, \tag{2}$$

where MOP (t) and MOA (t) are values at tth iteration, and t and T represent the current iteration and the maximum iteration, respectively. Max and Min represent, respectively, the maximum and minimum values of the accelerated function, and α is a sensitive parameter and represents the accuracy of exploitation in the whole iterative process.

In the stage of exploration, the division (D) and the multiplication (M) have high dispersion that probably leads to divergence. However, the communication between operators is increased to support the search in the exploration phase after several iterations. The position updates in the exploring phase are defined as

$$x_{i,j}(t+1) = \begin{cases} \text{best}(x_j) \div (\text{MOP} + \varepsilon) \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & r_2 < 0.5, \\ \text{best}(x_j) \times \text{MOP} \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & \text{otherwise}, \end{cases} \tag{3}$$

where $x_{i,j}(t)$ represents the jth position of the ith solution at current iteration, best (x_j) is the best-obtained solution in the jth position so far, ε is a small integer number, UB_j and LB_j represent the upper value and lower bound value, respectively, in the jth position, μ is a control parameter to adjust search process and is set equal to 0.499, and r_2 is a random number between 0 and 1.

In the stage of exploiting, the subtraction (S) or addition (A) has low dispersion, which helps them approach the optimal solution. The position updating equations are proposed for the exploitation parts as

$$x_{i,j}(t+1) = \begin{cases} \text{best}(x_j) - \text{MOP} \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & r_3 < 0.5 \\ \text{best}(x_j) + \text{MOP} \times \left[\left(\text{UB}_j - \text{LB}_j\right) \times \mu + \text{LB}_j\right], & \text{otherwise}, \end{cases} \tag{4}$$

where r_3 is a random number between 0 and 1 and other parameters are set as in equation (3).

2.2. Robot Path Planning. Autonomous navigation is one of the most important issues of intelligence. Robot path planning is designed for the target of avoiding barriers in the procedure of navigation. The navigation consists of four essential requirements known as perception, localization, cognition, and planning. Motion control in path planning is the most important part [36]. Normally, there are many feasible paths for a robot to reach the target from the starting position. However, in actual situations, the best feasible path is selected based on the shortest distance, path smoothness, minimum energy consumption, and the like [37] The path planning strategy of mobile robots can be categorised as classical methods and heuristic methods. However, classical methods do not always find the optimal path and are often locked in some local optimum. In addition, in the presence of multiple barriers or dynamic environments, some of them may not provide suitable solutions. To avoid the limitations of classical methods, heuristic methods is employed [38].

In this article, the improved AOA is applied to the robot path planning based on the position of static barriers. The mathematical model of robot path planning will be introduced in Section 5.1.

3. Improved AOA

3.1. Adaptive AOA. There are two hyperparameters in AOA: α and μ, where μ controls the absolute step size of the algorithm position update. Under a given objective function, it is a fixed value and will not change with the number of iterations. However, α is used to calculate MOP, and MOP is used to control the magnitude of the change of AOA in the iterative process. Experiments have shown that different α values will affect MOP and thus the performance of the algorithm [28]. Through experimental comparison, the results show that when $\alpha < 1$, MOP is a convex function, which is conducive to the full exploration of the algorithm in the early stage, rather than quickly converging to a local area. Based on these findings, we propose an adaptive change of α. A small value of α may cause the algorithm falling into a local optimum, and a large value of α may lead to insufficient search and even cannot find the optimal solution, which will affect the algorithm's efficiency. Therefore, it is useful to introduce an adaptively changed α to improve the balance between the exploration phase and exploitation phase's search capabilities of the algorithm. In this article, we change the size of α according to the fitness value and proposes a formula such as equations (5) and (6).

$$\alpha'(t) = \begin{cases} \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \times \dfrac{f - f_{\min}}{f_{\text{avg}} - f_{\min}}, & f \le f_{\text{avg}}, \\ \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \times \dfrac{f_{\max} - f}{f_{\max} - f_{\text{avg}}}, & f > f_{\text{avg}}, \end{cases} \tag{5}$$

$$\alpha(t) = 1 - \alpha'(t) + \varepsilon, \tag{6}$$

where $\alpha(t)$ is the value of α at the tth iteration, $\alpha_{\min}$ and $\alpha_{\max}$ are the minimum value and the maximum value of α according to the experience value, respectively; f, $f_{\min}$, $f_{\max}$, f_{avg} are the fitness value, the minimum fitness value, the maximum fitness value, and the average fitness value, at

the current iteration, respectively; and ε is a very small positive integer. The value of α' is related to the value of α, which is inversely proportional, and the values of $f - f_{\min}/f_{\text{avg}} - f_{\min}$ and $f_{\max} - f/f_{\max} - f_{\text{avg}}$ are proportional to α'. It is better for the values of $f - f_{\min}/f_{\text{avg}} - f_{\min}$ and $f_{\max} - f/f_{\max} - f_{\text{avg}}$ be closer to 1. Therefore, we replace $f_{\max} - f_{\min}$ with $f_{\text{avg}} - f_{\min}$ and $f_{\max} - f_{\text{avg}}$ in equations (5) and (6).

3.2. Parallel Strategy. In order to improve computing speed and solve large and complex problems, parallel strategy is widely used, such as Data Mining [39] and Deep Learning [40]. In the field of intelligent algorithms, GA is one of the earliest algorithms employed parallel strategy [32]. Experiments have proved that adding a parallel strategy can implement multipoint parallel search in space, which increases the communication between populations. Due to the use of multiple CPUs, the search speed is accelerated. Therefore, the algorithm with the parallel strategy performs better than the original algorithm [31–35]. With the increment of data, a single communication strategy is usually not adequate. Therefore, Roddick [41] expanded the single communication strategy into three communication strategies and applied them to PSO. Experiments proved that the three communication strategies successfully increased the efficiency of PSO. Then, many researchers improve the parallel method based on Chang's theory. For example, Yang et al. [31] and Zhu et al. [19] proposed a parallel strategy of multiple groups and multiple strategies. Each population will update according to its own communication strategy. When a certain number of iterations are reached, the populations communicate with each other and exchange information. Their final update methods are the same, which use the optimal value of one group instead of the worst value of another group. Nasrabadi et al. [35] adopted the parallel strategy of multiple groups of the same strategy. At the beginning, each group used the same strategy to evolve independently. The populations reach a certain number of iterations and began to exchange information.

In this article, we add randomness to the communication strategy and use different strategies in local search and global search, which can help groups fully communicate, as shown in Figure 1. Specifically, for the local: two groups are arbitrarily selected, and one group of particles with the best fitness is substituted for the other group of particles with the worst fitness every T iteration. For the global, the global best particle replaces the worst particle in the group every $2T$ iterations. The purpose of using these two strategies is to enhance the randomness of the algorithm, strengthen the communication between populations, and avoid premature convergence of the algorithm, thereby improving the robustness of the algorithm.

Following to the above introduction, Figure 2 shows the model of adaptive parallel AOA (APAOA) search process. It can be seen that the optimal solution of the algorithm is first updated by the multiplication and division operator in the exploration stage, then updated by the addition and subtraction operator in the exploitation stage and finally find the global optimal solution. The pseudocode of the APAOA algorithm is shown in Algorithm 1.

4. Experimental Analysis of the Algorithm

In this section, the experimental research will be committed. Section 4.1 introduces 18 benchmark functions provided by [42], including unimodal functions, multimodal function, and fixed-dimensional multimodal functions. Section 4.2 tests APAOA and original AOA in 30 dimensions (30D). Section 4.3 compares APAOA with other popular intelligent algorithms on 100 dimensions (100D).

The algorithms are compared using mean, standard deviation, and Friedman ranking (Rank) test. These are popular evaluation indexes in algorithm comparisons.

4.1. Benchmark Functions. In this article, 18 benchmark functions are employed as one of the main methods to test the performance of intelligent algorithms. Benchmark functions are shown in Tables 1–3.

4.2. Comparison with the Original AOA and APAOA with 30D. To verify the performance of the APAOA algorithm, it is compared with the original AOA on unimodal functions, multimodal functions, and fixed-dimensional multimodal function. AOA is proved to perform the best on 30D in the study by Abualigah et al. [28]. The dimensions in this experiment are set to 30, the number of populations is set to 40, the maximum number of iterations is 500, and populations in APAOA with parallel strategy are divided into four groups, and each algorithm is run independently for 30 times. In Tables 4–6, the Ave represents the mean value, the Std indicates the standard deviation, and the Best represents the optimal value. The bold part of the tables indicate victory.

According to the results in Table 4, compared with AOA, APAOA has advantages in 5 of the 7 unimodal functions. Among them, APAOA is not as good as AOA in terms of mean value or optimal value in test functions F5 and F6, but the gap between the two algorithms is close, and APAOA is better than AOA in standard deviation in test function F5. This shows that the dispersion degree of solution accuracy of APAOA on F5 tends to be stable.

Figure 3 shows the convergence speed and accuracy of the two algorithms on the test function F1–F7. APAOA is better than AOA on the whole, except for the poor optimization ability of F5 and F6. In the unimodal test function of AOA, only F4 and F7 can converge. However, APAOA can converge quickly and find better values than AOA on the other six test functions except test function F5.

On the six multimodal test functions, APAOA is better than AOA on F8, F9, F10, and F12, and the solution accuracy is 10 times higher than the original, such as F9 and F10. APAOA is only worse than AOA on F11. On the test function F13, the two algorithms are almost the same in mean, standard deviation, and optimal value; the value of AOA is more stable than that of APAOA, and APAOA can find a better solution than AOA. The test results are shown in Table 5.

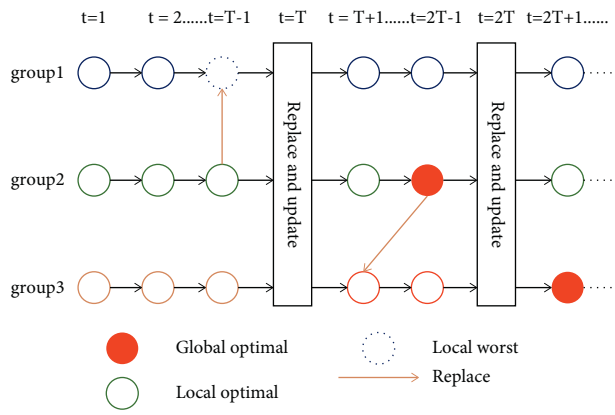


Figure 1: Parallel communication strategy process.

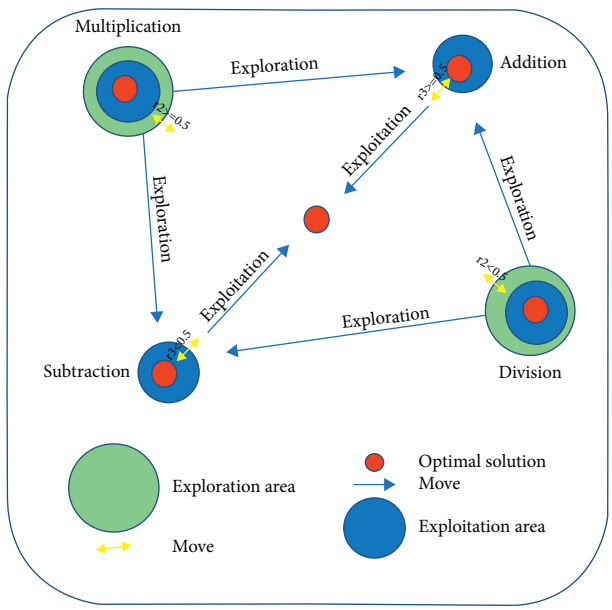


Figure 2: APAOA search process model.

According to Figure 4 and Table 5, on F13, neither algorithm can converge. On the other five test functions, APAOA can converge quickly and find the better solution. However, APAOA is not as good as AOA in all aspects on F11. This shows that APAOA still has the problem of insufficient accuracy of solution.

The benchmark functions F14–F18 are fixed-dimensional multimodal functions and low dimensional. It can be found that the performance of the two algorithms is close according to the results in Table 6. APAOA has superior performance on F14 and F18, whereas the performance on F15 and F17 is not as good as AOA. On F16, the performance of the two algorithms is almost the same. From Figure 5, both algorithms can converge, but the convergence speed of APAOA is still faster than that of AOA on the whole, except for F15.

To sum up, the APAOA we proposed, which adaptively changes the parameter α, can make algorithm jump out of the local optimal solution such as F5 and F13, and the advantage of parallel strategy is that the algorithm can quickly converge and find the better solution in the face of complex and high-dimensional environment, such as F1–F4. However, it also performs not good in some functions such as F11 and F15.

4.3. Comparisons with Popular Algorithm. In Section 4.2, APAOA has been compared with AOA in unimodal function, multimodal function, and fixed-dimensional multimodal function. Because the dimension of test function F14–F18 is fixed, the dimension of F1–F13 can be expanded. In this section, in order to further evaluate the ability of APAOA to solve high-dimensional problems, the APAOA was compared with popular optimization algorithms on F1–F13. Table 7 shows popular algorithms and their parameter settings. To achieve a fair comparison, all algorithms are tested on 100D, and the common parameters settings are the same as those in section 4.2.

```
Initialize the AOA parameters μ, groups;
Initialize the candidate solutions randomly (candidate solutions: i = 1, . . ., N);
while (t < T)
  Calculate the fitness function for the given solutions;
  Find the best solution so far;
  for g = 1: groups
    if rand <0.5
      Perform parallel communication strategy 1 every T iteration;
    else
      Perform parallel communication strategy 2 every 2T iteration;
    end if
    Update the α value using equations (5) and (6);
    Update the MOA value using equation (1);
    Update the MOP value using equation (2);
    for i = 1: N
      for j = 1: dimensions
        Generate random values between [0, 1];
        if r1 < MOA//enter in exploration phase
          Update the solutions by equation (3);
        else//enter in exploiting phase
          Update the solutions by equation (4);
        end if
      end for
    end for
  end for
  t = t + 1;
end while
Return the best solution.
```

Algorithm 1: Pseudocode of APAOA.

Table 1: Unimodal test functions.

Name	Function	Dim	Range	f min
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
F2	$f(x) = \sum_{i=0}^{n} \lvert x_i \rvert + \prod_{i=0}^{n} \lvert x_i \rvert$	30	[−10, 10]	0
F3	$f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} x_j)^2$	30	[−100, 100]	0
F4	$f(x) = \max_i \{\lvert x_i \rvert, \quad 1 \le i \le n\}$	30	[−100, 100]	0
F5	$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	30	[−30, 30]	0
F6	$f(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100, 100]	0
F7	$f(x) = \sum_{i=0}^{n} i x_i^4 + \text{random}[0, 1)$	30	[−128, 128]	0

The experimental results in Table 8 show that the APAOA has achieved good results on the test functions F1–F13, in which the dimension is set to 100, and it is in the first place in the Friedman ranking test. On F1–F13, APAOA achieved the best performance except that F6 and F11 did not perform as expected. The performance of APAOA on F6 is second only to SSA, but there is a large gap between them. APAOA only performed not good in F11. In the whole, APAOA outperforms other popular algorithm on most functions.

Next, the convergence and search ability of different algorithms are evaluated, and the results are shown in Figure 6. According to Table 8, it is found that there is a big gap between the search ability of other popular algorithms and APAOA. If all convergence curves are drawn on one graph, the convergence curve of APAOA will be seen like a straight line. Therefore, the convergence curve of APAOA is drawn separately (right of Figure 6) in this article. Among the 13 test functions, APAOA has poor convergence ability only on F8, F11 and F13, but its ability to search the optimal solution is stronger than other algorithms. On F2 and F4, APAOA converges extremely well and can find better solutions than other algorithms. On F2, the search capabilities of all algorithms differ greatly, causing the algorithm's convergence curve to become a straight line.

5. Application in Robot Path Planning

5.1. Robot Path Planning Mathematical Model. The robot path planning problem generally include three aspects: environment modelling, path searching, and path smoothing.

Table 2: Multimodal test functions.

Name	Function	Dim	Range	f min
F8	$f(x) = \sum_{i=1}^{n} [-x_i \sin(\sqrt{\lvert x_i \rvert})]$	30	[−500, 500]	$-418.9829 \times n$
F9	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
F10	$f(x) = -20\exp(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2} - \exp 1/n\sum_{i=1}^{n}\cos(2\pi x_i) + 20 + e)$	30	[−32, 32]	0
F11	$f(x) = 1 + 1/4000\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}}$	30	[−600, 600]	0
F12	$f(x) = x/n\{10\sin(\pi y_1)\} + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}) + \sum_{i=1}^{n} u(x_i, 10, 100, 4)]$, where $y_i = 1 + x_i + 1/4$, $u(x_i, a, k, m)\begin{cases} K(x_i - a)^m, & \text{if } x_i > a, \\ 0, & -a \le x_i \ge a, \\ K(-x_i - a)^m, & -a \le x_i \end{cases}$	30	[−50, 50]	0
F13	$f(x) = 0.1(\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 1 + \sin^2(2\pi x_n)) + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50, 50]	0

Table 3: Fixed-dimension multimodal test function.

Name	Function	Dim	Range	f min
F14	$f(x) = [1/500 + \sum_{j=1}^{25} 1/j + \sum_{i=1}^{2} (x_i - a_{ij})]^{-1}$	2	$[-65, 65]$	1
F15	$f(x) = \sum_{i=1}^{11} [a_i - x_1 (b_i^2 + b_i x_2)/b_i^2 + b_i x_3 + x_4]^2$	4	$[-5, 5]$	0.00030
F16	$f(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	−1.0316
F17	$f(x) = (x_2 - 5.1/4\pi^2 x_1^2 + 5/\pi x_1 - 6)^2 + 10(1 - 1/8\pi)\cos x_1 + 10$	2	$[-5, 5]$	0.398
F18	$f(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_i + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	$[-2, 2]$	3

Table 4: Unimodal function test on 30D.

Algorithm	AOA			APAOA		
F	Ave	Std	Best	Ave	Std	Best
F1	$2.95e-06$	$1.10e-06$	$1.53e-06$	$\mathbf{3.72e-07}$	$2.52e-07$	$4.74e-09$
F2	$1.29e-03$	$1.38e-03$	$2.83e-05$	$\mathbf{5.84e-04}$	$2.81e-03$	$8.05e-06$
F3	$6.07e-04$	$5.38e-04$	$1.24e-05$	$\mathbf{2.05e-04}$	$4.25e-04$	$1.26e-07$
F4	$1.42e-02$	$1.04e-02$	$0.30e-02$	$\mathbf{1.33e-02}$	$3.31e-02$	$1.03e-05$
F5	$\mathbf{2.77e+01}$	$2.47e-01$	$2.81e+01$	$2.87+01$	$1.35e-01$	$2.85+01$
F6	$\mathbf{2.56e+00}$	$2.10e-01$	$2.34e+00$	$3.81e+00$	$1.22e+00$	$2.64e-01$
F7	$5.89e-05$	$5.87e-05$	$9.58e-05$	$\mathbf{7.38e-06}$	$8.00e-06$	$3.32e-06$

Table 5: Multimodal function test on 30D.

Algorithm	AOA			APAOA		
F	Ave	Std	Best	Ave	Std	Best
F8	$-5.49e+03$	$3.26e+02$	$-5.35+03$	$\mathbf{-5.25e+03}$	$1.72e+02$	$-8.22e+03$
F9	$8.15e-07$	$7.52e-07$	$1.46e-06$	$\mathbf{1.91e-09}$	$9.18e-08$	$2.26e-10$
F10	$2.92e-04$	$1.65e-04$	$4.23e-04$	$\mathbf{4.29e-05}$	$7.51e-05$	$1.27e-05$
F11	$\mathbf{1.25e-03}$	$4.69e-03$	$2.09e-05$	$4.81e+01$	$1.24e+01$	$1.60e+02$
F12	$6.89e-01$	$2.86e-02$	$0.71e+00$	$\mathbf{5.10e-01}$	$3.40e-01$	$0.51e+00$
F13	$2.96e+00$	$2.63e-02$	$2.97e+00$	$2.96e+00$	$2.67e-02$	$2.87e+00$

Table 6: Fixed-dimensional multimodal function test.

Algorithm	AOA			APAOA		
F	Ave	Std	Best	Ave	Std	Best
F14	$1.05e+01$	$3.50e+00$	$5.93e+00$	$\mathbf{1.00e+01}$	$5.82e+00$	$9.98e-01$
F15	$\mathbf{7.02e-03}$	$1.03e-02$	$1.31e-02$	$1.22e-02$	$1.27e-02$	$1.10e-03$
F16	$-1.03e+00$	$2.25e-11$	$-1.03e+00$	$-1.03e+00$	$1.33e-02$	$-1.03e+00$
F17	$\mathbf{3.98e-01}$	$2.05e-06$	$4.00e-01$	$4.76e-01$	$1.67e-01$	$4.00e-01$
F18	$9.30e+00$	$1.16e+01$	$3.00e+00$	$\mathbf{3.00e+00}$	$2.03e-03$	$3.00e+00$

5.1.1. Environment Modelling. In an actual working environment, a robot has to face the complexity and quick change. To regularize the experimental environment, we use a grid-like method for modelling. Figure 7 shows the robot workplace model. The robot path planning problem is transformed into the use of algorithms to make the robot walk from the source to target, avoiding collisions with barriers during the procedure and finding an optimal collision-free path.

5.1.2. Path Searching. On the basis of environmental modelling, the issue of finding an optimal path is transferred into the issue of an objective function obtaining the optimal value. Assuming that in an ideal state, the robot is a straight collision-free path from the source to the target, and the objective function is shown as

$$D = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2}, \tag{7}$$

where D represents the distance between the source and the target, (x_s, y_s) represents the source, and (x_t, y_t) represents the target. However, in the actual working environment, the robot will encounter barriers to prevent it from moving forward. Firstly, for the robot workplace model, a Cartesian coordinate system is constructed, and the horizontal axis and vertical axis of the two-dimensional plane are equidistantly divided to form a set of intersections Node $= \{(x_i, y_i), \quad i = 1, 2, 3, \ldots, n\}$, the sum of the distances between the particles randomly generated by the algorithm is L and is given as:

$$L = \sum_{j=1}^{k} \sqrt{\left(x_{j+1} - x_j\right)^2 + \left(y_{j+1} - y_j\right)^2}, \quad (k < n). \tag{8}$$

Secondly, we use equation (9) to search for points in the barriers on the working path of the robot. If any point on the path is within the barrier, the penalty function equation (10) is added to the objective function.

$$d = \frac{\sqrt{(x_i - x_b)^2 + (y_i - y_b)^2}}{r_b} - 1, \quad \begin{cases} d > 0, & \text{not in barrier,} \\ d < 0, & \text{in barrier,} \end{cases} \tag{9}$$

where (x_b, y_b) is the coordinate of the centre point of the obstacle b, r_b is the radius of the obstacle, and $b = 1, 2, 3, \ldots$.

$$\text{Penal} = \text{Penal} + \text{mean}\,(\min(d, 0)), \tag{10}$$

where Penal is the penalty function.

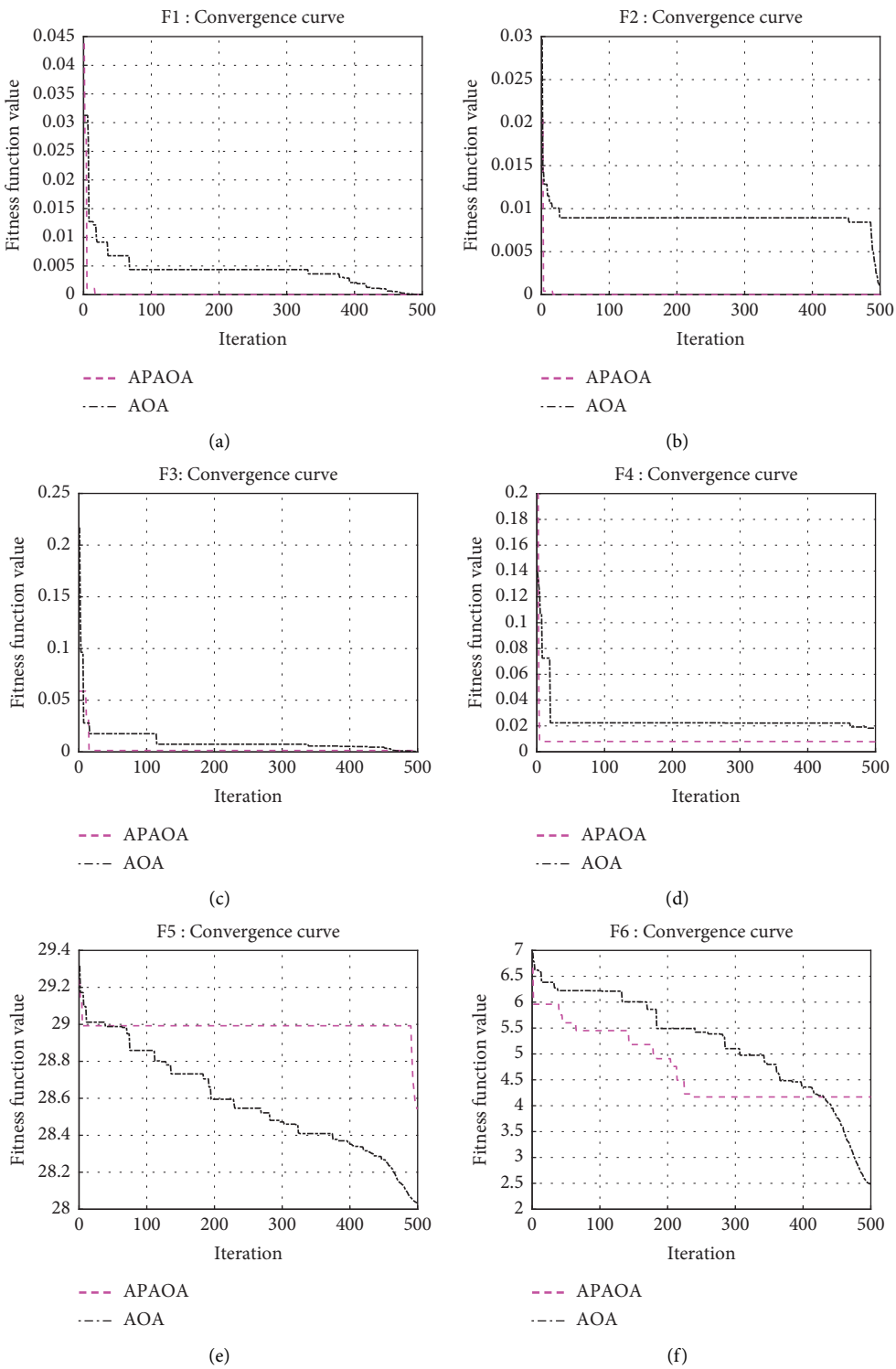
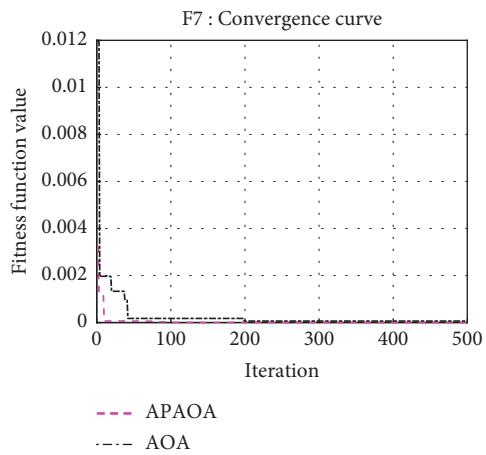


Figure 3: Continued.



(g)

Figure 3: Comparison of convergence curves between APAOA and AOA in F1–F7.

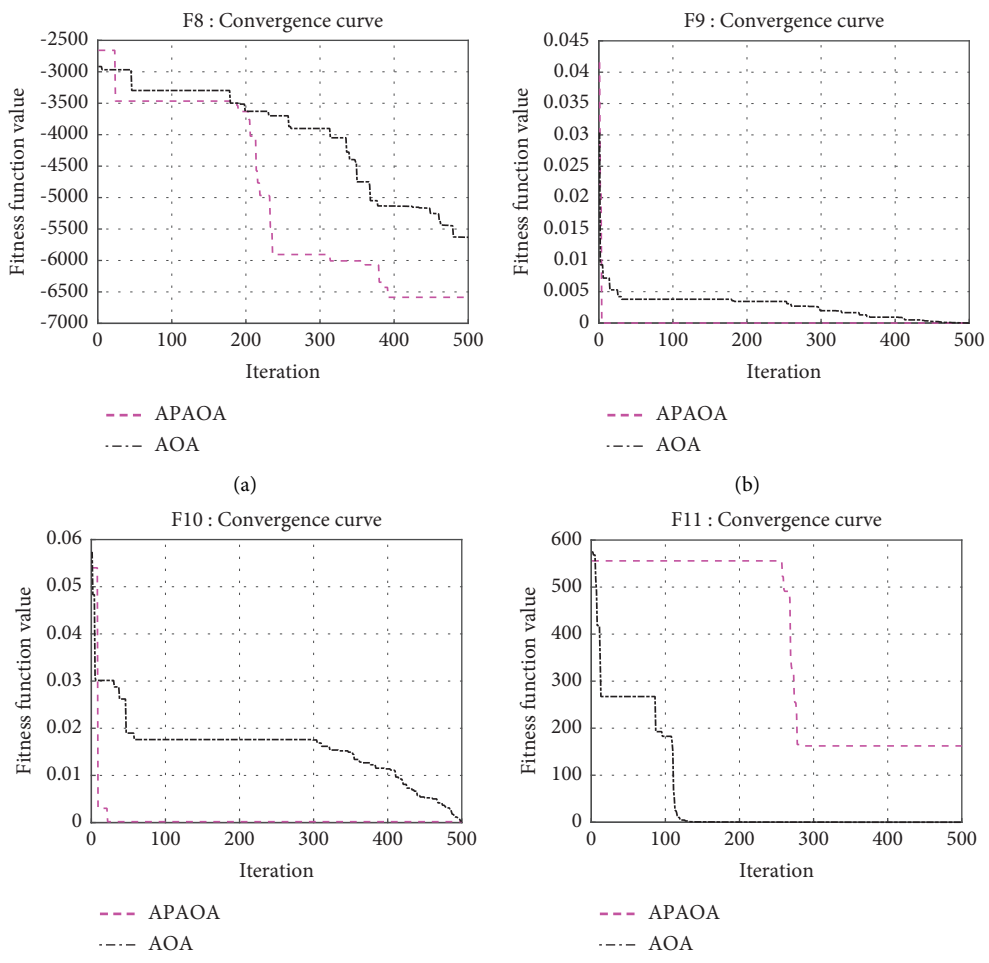


Figure 4: Continued.

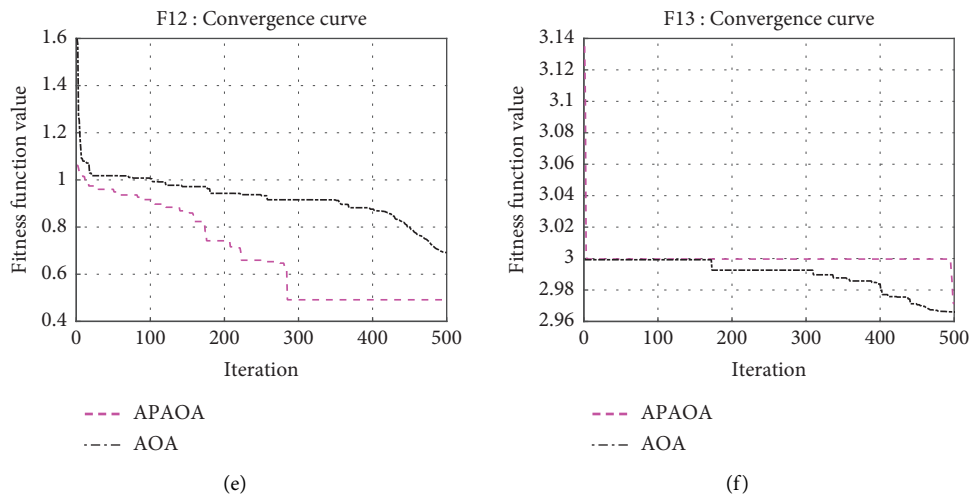


Figure 4: Comparison of convergence curves between APAOA and AOA in F8–F13.

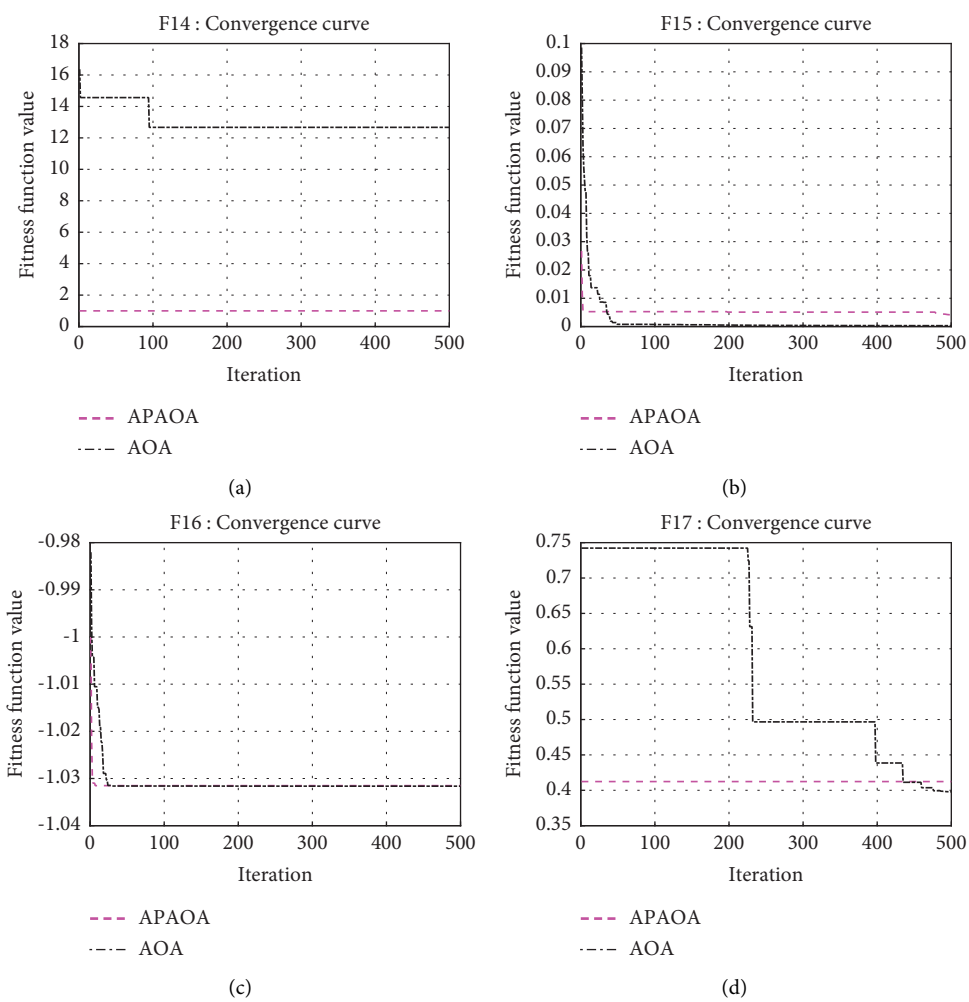
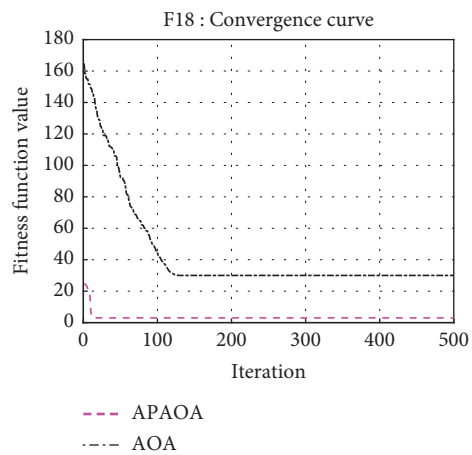


Figure 5: Continued.

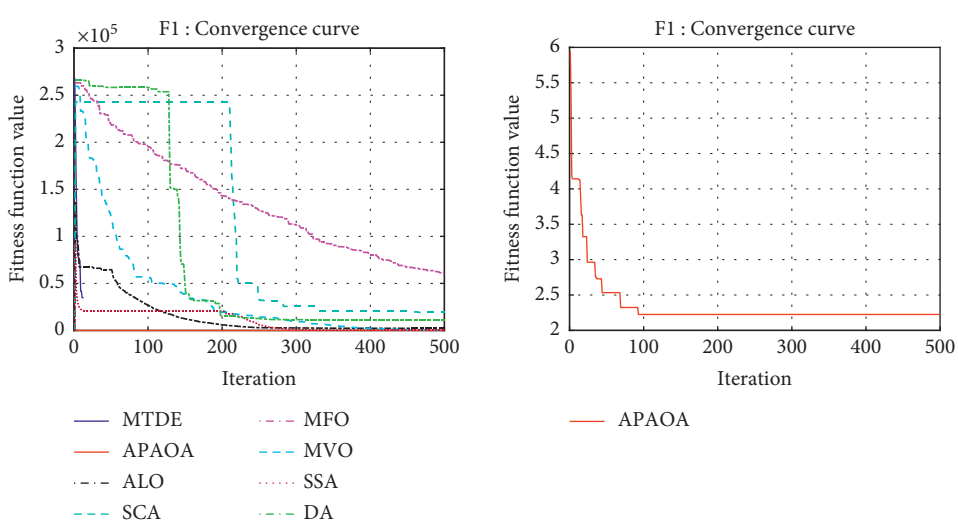


(e)

Figure 5: Comparison of convergence curves between APAOA and AOA in F14–F18.

Table 7: Parameter settings for algorithms.

Algorithm	Parameter value
ALO	None
MVO	$\mathrm{WEP}_{\mathrm{Max}} = 1$; $\mathrm{WEP}_{\mathrm{Min}} = 0.2$; $p = 6$
SCA	$a = 2$
MFO	$b = 1$, $a = [-2, -1]$ (linearly decrease)
DA	$r, \varepsilon\_\max = (\mathrm{ub} - \mathrm{lb})/10$; $(\mathrm{ub}, \mathrm{lb})$ is maximum and minimum boundary
SSA	$c_1 = [0, 2]$ (linearly reduce)
MTDE [43]	$F = 0.5$; $\mathrm{CR} = 0.8$
APAOA	$\mu = 0.499$; $\alpha_{\max} = 1$; $\alpha_{\min} = 0.1$
PSO	$c_1 = c_2 = 1.5$, $w = 1$
GA	$\mathrm{pc} = 0.8$, $\mathrm{pm} = 0.2$
ACO	$\alpha = 1$, $\beta = 7$, $\mathrm{Rho} = 0.2$, $Q = 1$



(a) (b)

Figure 6: Continued.

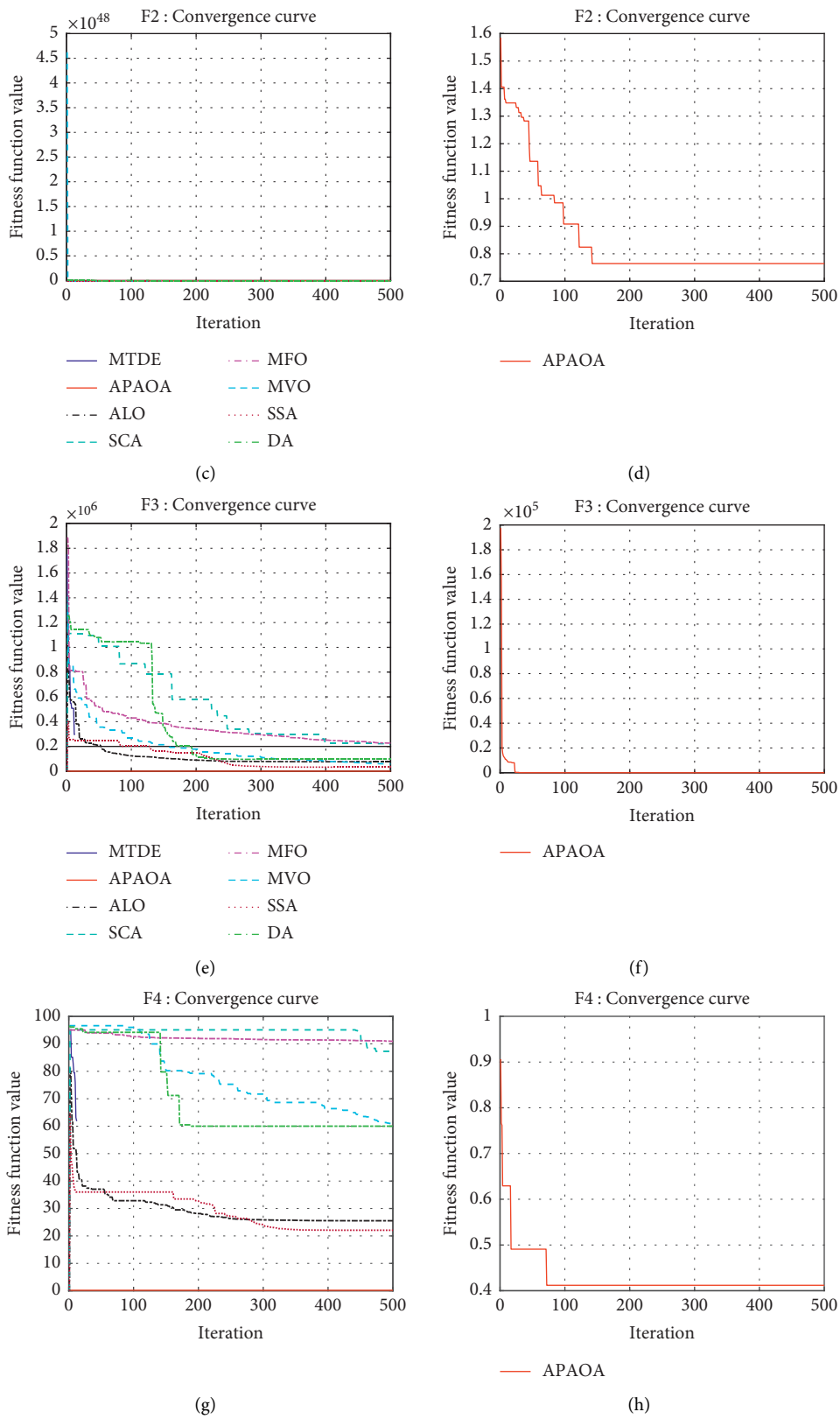


Figure 6: Continued.

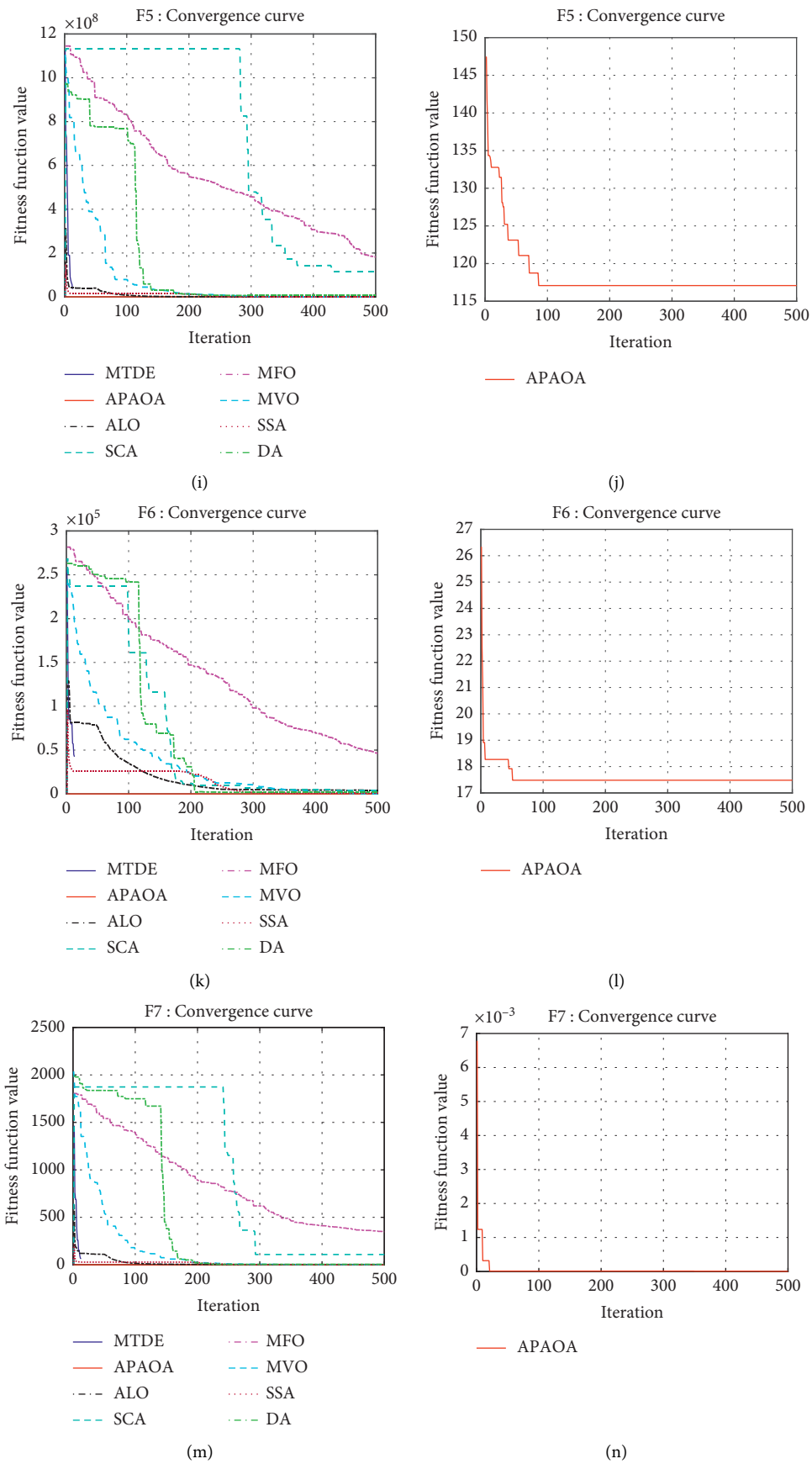


Figure 6: Continued.

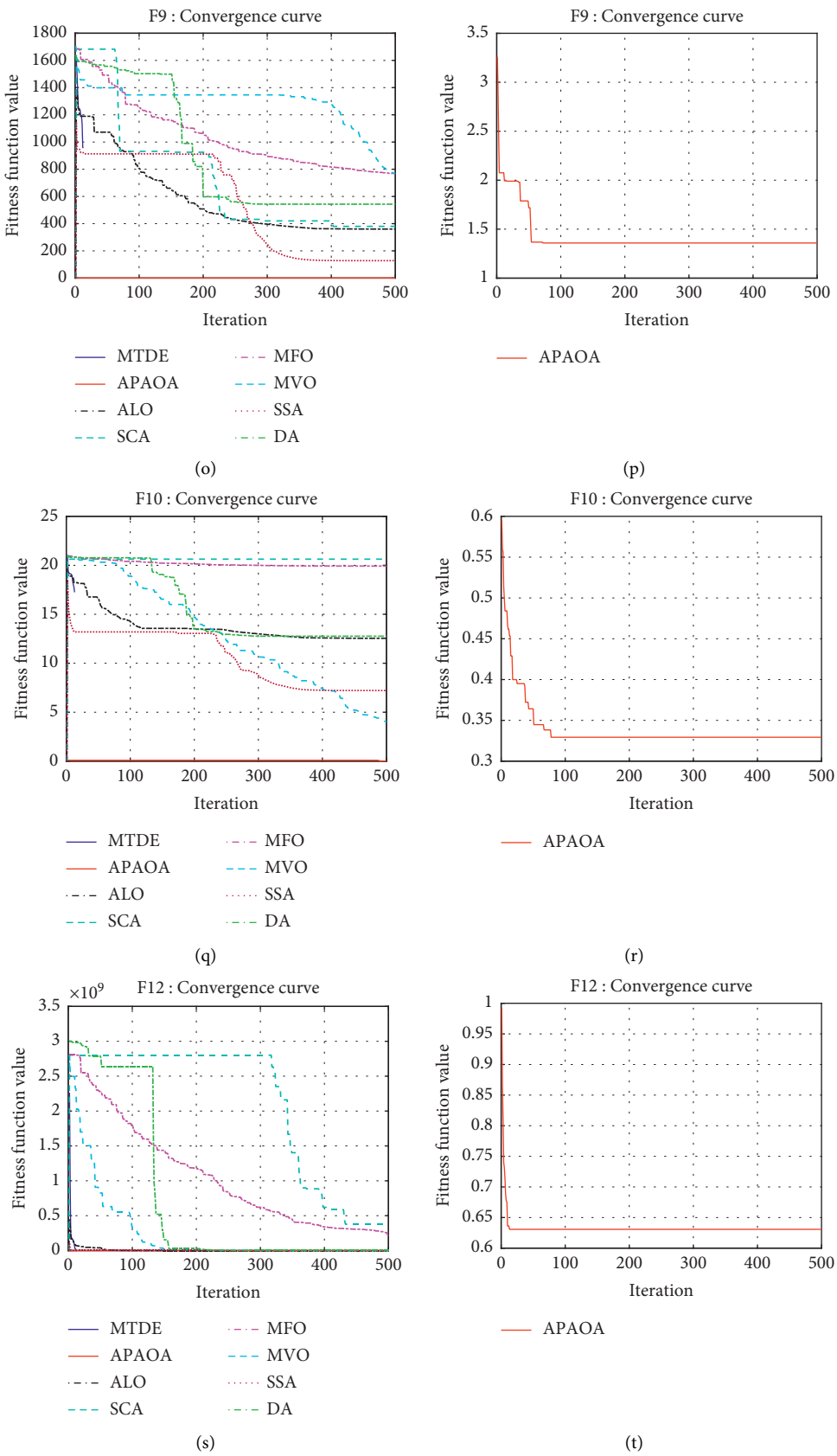


Figure 6: Continued.

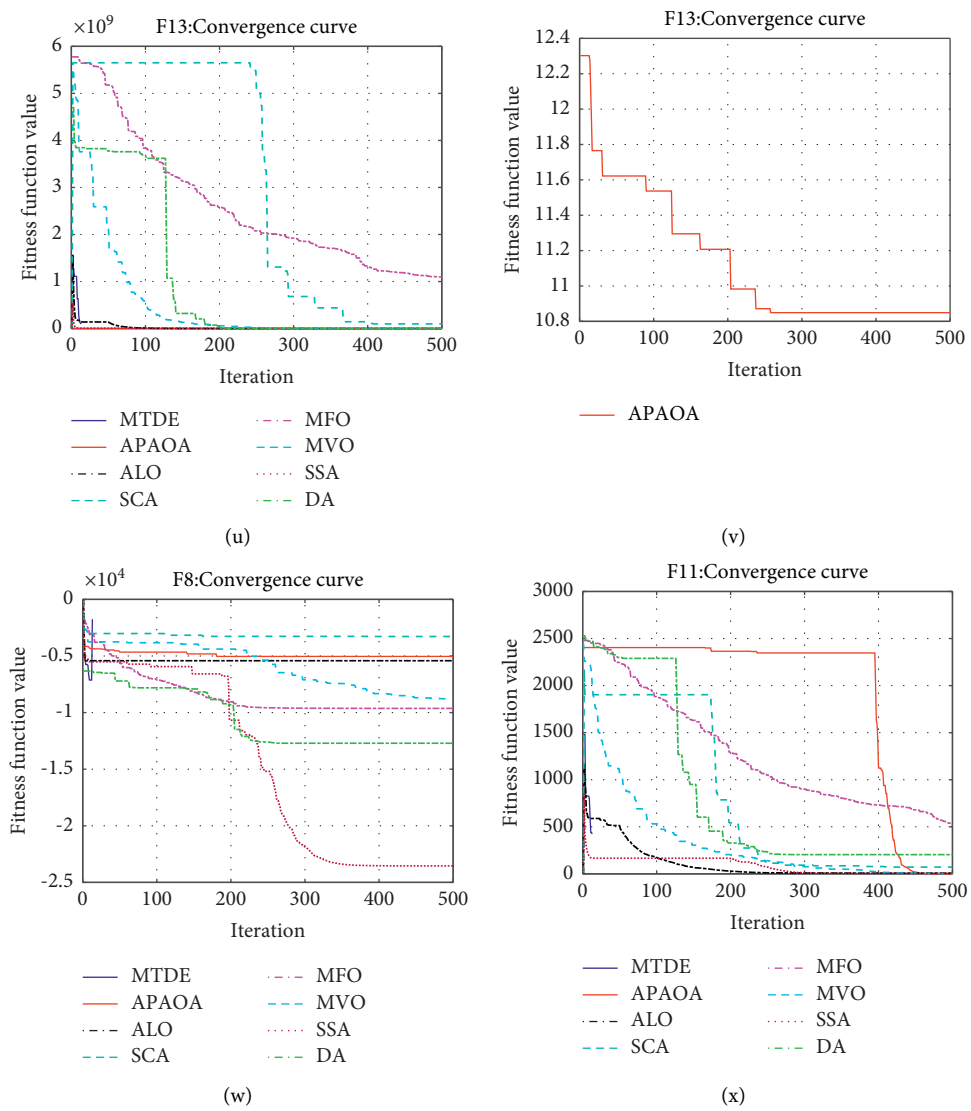


Figure 6: Comparison results of convergence performance of popular optimization algorithms.

In summary, the objective function of path planning is shown as equation (11), and the purpose is to use the penalty function to avoid barriers and achieve the shortest path, which is the optimal value.

$$\mathrm{Rp} = L * (1 + \omega * \mathrm{Penal}), \tag{11}$$

where ω is a penalty factor, which is verified by experiments that 10 is a suitable value. Therefore, it is set to 10 in this article.

5.1.3. Path Smoothing. Finally, because the connection between points is a straight line, and the path that the algorithm planned is not a feasible path, so we use Spline interpolation to smooth the solution to achieve a better accuracy.

5.2. Simulation Research. For the robot path planning mentioned in Section 5.1, we have meshed the environment and modelled the barriers. We set the two-dimensional space to be 6×6. The source coordinates and target coordinates of the two scenes are the same, which are (0, 0) and (4, 6).

To further verify the performance of the APAOA, we choose the classical algorithms, such as PSO, ACO, and GA to compare with APAOA and AOA. During the simulation, the algorithm parameters are shown in Table 7, and each algorithm runs independently 30 times. To compare the performances of the algorithms in different environments, two test environments are committed, with the number of barriers being 4 and 7. The experimental results are shown in Figures 8 and 9 and Tables 9 and 10. Tables 9 and 10 show

Table 8: Comparison results of popular algorithms in F1–F13.

F	ALO			SSA			MVO		
	Ave	Std	Rank	Ave	Std	Rank	Ave	Std	Rank
F1	$1.26e+03$	$6.29e+02$	**4**	$3.66e+00$	$1.69e+00$	**2**	$1.20e+02$	$2.38e+01$	**3**
F2	$2.89e+02$	$1.86e+02$	**6**	$1.14e+01$	$2.76e+00$	**3**	$5.65e+22$	$3.02e+23$	**7**
F3	$5.93e+04$	$2.17e+04$	**3**	$9.40e+03$	$3.66e+03$	**2**	$5.69e+04$	$5.91e+04$	**4**
F4	$3.13e+01$	$4.59e+00$	**3**	$1.26e+01$	$1.32e+00$	**2**	$5.46e+01$	$5.87e+00$	**5**
F5	$3.02e+05$	$2.23e+05$	**4**	$1.16e+03$	$1.14e+03$	**2**	$7.01e+03$	$7.12e+03$	**3**
F6	$1.68e+03$	$5.16e+02$	**4**	$3.22e+00$	$1.45e+00$	**1**	$1.15e+02$	$1.99e+01$	**3**
F7	$3.30e+00$	$1.02e+00$	**4**	$4.00e-01$	$8.23e-02$	**2**	$5.06e-01$	$1.05e-01$	**3**
F8	$-1.81e+04$	$3.83e-12$	**8**	$-2.43e+04$	$1.699e+03$	**2**	$-2.33e+04$	$1.32e+03$	**4**
F9	$2.98e+02$	$5.35e+01$	**4**	$9.69e+01$	$2.74e+01$	**2**	$6.77e+02$	$6.71e+01$	**5**
F10	$1.28e+01$	$1.44e+00$	**5**	$4.36e+00$	$6.58e-01$	**2**	$8.22e+00$	$6.69e+00$	**3**
F11	$1.43e+01$	$6.34e+00$	**3**	$7.76e-01$	$1.97e-01$	**1**	$2.07e+00$	$1.75e-01$	**2**
F12	$1.45e+02$	$2.70e+02$	**4**	$5.16e+00$	$1.45e+00$	**3**	$1.81e+00$	$5.79e+00$	**2**
F13	$1.63e+04$	$3.03e+04$	**4**	$1.32e+02$	$2.08e+01$	**2**	$1.47e+02$	$2.78e+01$	**3**
Mean		**4.31**			**2.00**			**3.62**	
Rank		**4**			**2**			**3**	

F	SCA			MFO			DA		
	Ave	Std	Rank	Ave	Std	Rank	Ave	Std	Rank
F1	$9.25e+03$	$5.25e+03$	**5**	$5.91e+04$	$1.54e+04$	**7**	$1.49e+04$	$1.07e+04$	**6**
F2	$8.01e+00$	$7.18e+00$	**2**	$2.45e+02$	$4.07e+01$	**5**	$5.36e+01$	$2.01e+01$	**4**
F3	$2.29e+05$	$4.42e+04$	**6**	$2.38e+05$	$6.24e+04$	**7**	$2.05e+05$	$7.48e+04$	**5**
F4	$8.85e+01$	$2.83e+00$	**6**	$9.25e+01$	$2.01e+00$	**7**	$5.27e+01$	$8.55e+01$	**4**
F5	$1.13e+08$	$6.02e+07$	**6**	$1.52e+08$	$7.33e+07$	**7**	$2.39e+07$	$2.10e+07$	**5**
F6	$1.22e+04$	$6.01e+03$	**5**	$5.46e+04$	$1.29e+04$	**7**	$1.55e+04$	$8.96e+03$	**6**
F7	$1.19e+02$	$6.64e+01$	**6**	$2.30e+02$	$1.30e+02$	**7**	$3.51e+01$	$3.67e+01$	**5**
F8	$-7.23e+03$	$6.41e+02$	**6**	$-2.38e+04$	$2.36e+03$	**3**	$-1.03e+04$	$9.74e+02$	**7**
F9	$2.57e+02$	$1.05e+02$	**3**	$8.28e+02$	$8.29e+01$	**7**	$7.51e+02$	$1.17e+02$	**6**
F10	$1.95e+01$	$2.86e+00$	**6**	$1.98e+01$	$1.94e-01$	**7**	$1.21e+01$	$2.47+00$	**4**
F11	$9.57e+01$	$3.69e+01$	**5**	$5.19e+02$	$1.25e+02$	**6**	$6.03e+01$	$4.28e+01$	**4**
F12	$2.56e+08$	$1.41e+08$	**7**	$2.55e+08$	$2.15e+08$	**6**	$1.39e+07$	$2.02e+07$	**5**
F13	$4.71e+08$	$2.49e+08$	**7**	$4.57e+08$	$2.20e+08$	**6**	$2.39e+07$	$1.69e+07$	**5**
Mean		**5.38**			**6.31**			**5.08**	
Rank		**6**			**7**			**5**	

F	APAOA			MTDE		
	Ave	Std	Rank	Ave	Std	Rank
F1	$1.24e-01$	$1.00e-01$	**1**	$4.00e+05$	$0.00e+00$	**8**
F2	$1.43e-01$	$3.02e-02$	**1**	$3.99e+62$	$3.68e+62$	**8**
F3	$8.92e-01$	$3.69e-01$	**1**	$2.21e+08$	$0.00e+00$	**8**
F4	$1.82e-01$	$1.47e-02$	**1**	$1.00e+02$	$0.00e+00$	**8**
F5	$9.98e+01$	$1.19e+00$	**1**	$5.59e+10$	$2.62e+09$	**8**
F6	$1.85e+01$	$2.74e+00$	**2**	$4.04e+05$	$0.00e+00$	**8**
F7	$6.15e-06$	$4.27e-06$	**1**	$6.63e+09$	$8.03e+08$	**8**
F8	$-5.91e+03$	$1.46e+03$	**1**	$-5.97e+03$	$2.18e+02$	**5**
F9	$3.90e-02$	$4.75e-02$	**1**	$8.66e+04$	$3.88e+02$	**8**
F10	$5.18e-02$	$2.59e-02$	**1**	$2.13e+01$	$9.05e-02$	**8**
F11	$2.13e+03$	$5.96e+02$	**8**	$9.70e+02$	$3.81e+01$	**7**
F12	$9.00e-01$	$2.10e-01$	**1**	$6.06e+10$	$3.63e+09$	**8**
F13	$1.01e+01$	$2.38e-01$	**1**	$8.24e+10$	$4.28e+09$	**8**
Mean		**1.62**			**7.69**	
Rank		**1**			**8**	

that the APAOA can find a shorter path than other algorithms, and the APAOA execution effect is more stable. Compared with AOA, the improved effect of APAOA is significant.

From the two path planning diagrams (Figures 8 and 9), it can be seen intuitively that the APAOA can always find the best path, while the other algorithms are not, and the convergence graph proves that the APAOA can find the

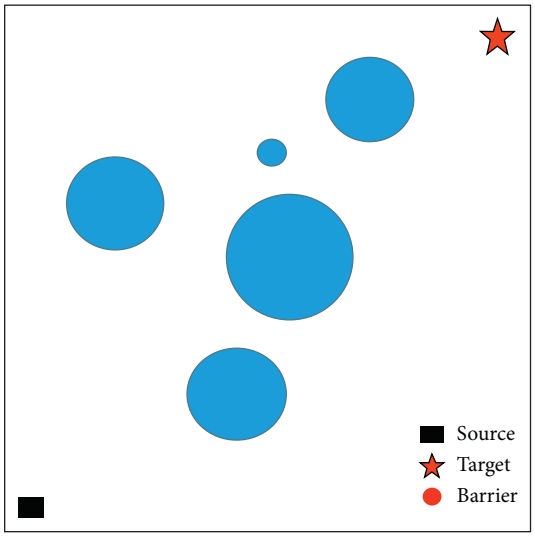
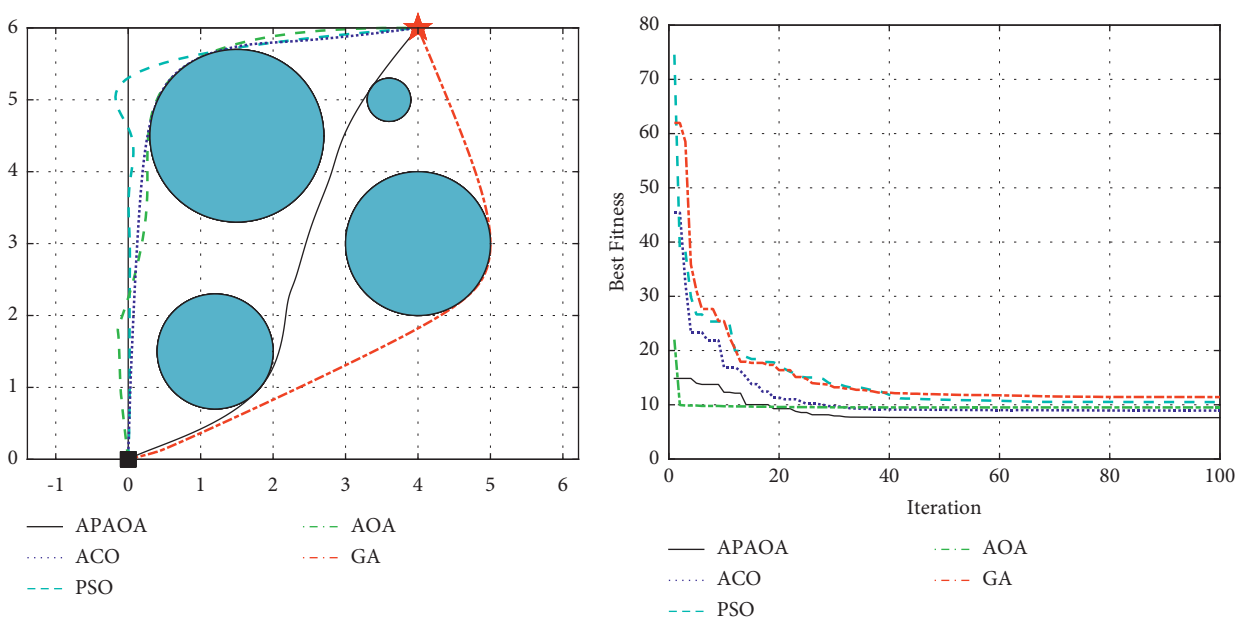


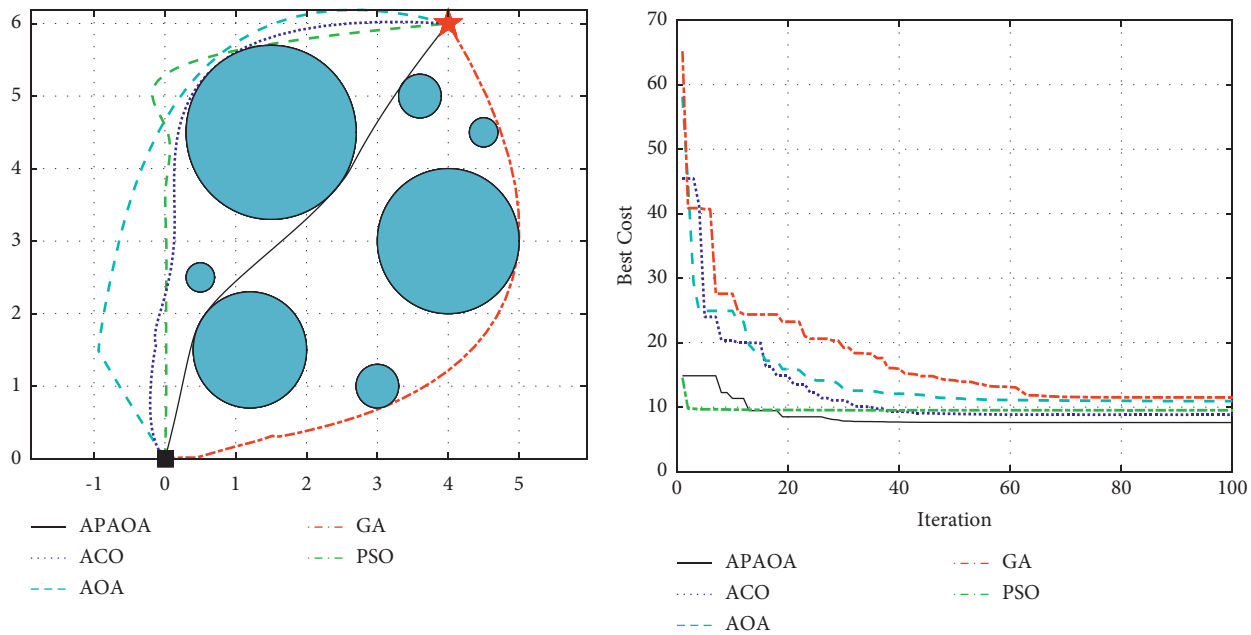
Figure 7: Robot workplace model.



(a) (b)

Figure 8: Path planning diagram (a) and convergence curve (b) of environment 1.

optimal path, and the convergence speed is fast. In summary, we propose the APAOA, which can better deal with the problem of robot path planning. However, there is still some problems such as insufficient accuracy and so on, which needs to be improved in the future to obtain better performance.



(a) (b)

Figure 9: Path planning diagram (a) and convergence curve (b) of environment 2.

Table 9: Comparison of the APAOA with the GA, PSO, ACO, and AOA in environment 1.

Algorithm	Mean	Best
PSO	9.32	9.08
ACO	8.32	8.10
GA	9.80	9.53
AOA	9.49	9.46
APAOA	**7.55**	**7.40**

Table 10: Comparison of the APAOA with the GA, PSO, ACO, and AOA in environment 2.

Algorithm	Mean	Best
PSO	10.39	9.79
ACO	8.06	7.65
GA	11.46	9.51
AOA	11.11	9.51
APAOA	**7.64**	**7.60**

6. Conclusions

In this article, an adaptive parallel AOA is proposed and applied to solve the problem of robot path planning. First of all, the adaptive equation proposed is established based on the fitness value of the particles. The benefits of the adaptive equation can enable the algorithm to better balance the search capabilities of the development and exploration phases and effectively avoid falling into local optimal solutions. The steps for the algorithm to enter the exploration and exploitation stage are adjusted. Experiments have shown that the algorithm performs better after the adjustment, and the solution accuracy is improved. Then, we also introduce a novel parallel strategy into the AOA algorithm to strengthen the communication among particles. By adopting the rule of "survival of the fittest," leaving elite individuals, it increases the robustness of the algorithm and achieves a significant improvement compared with the original algorithm. Finally, the robot path planning problem proposed in this article is used to further evaluate the performance of the algorithm. Compared with other algorithms, APAOA achieved better performance.

Data Availability

The data used to support this study are included within the article and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] H. Sarabu, K. Ahlin, and A. Hu, "Graph-based cooperative robot path planning in agricultural environments," in *Proceedings of the 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 519–525, Hong Kong, China, July 2019.

[2] S. Sehestedt, S. Kodagoda, and G. Dissanayake, "Robot path planning in a social context," in *Proceedings of the 2010 IEEE Conference on Robotics, Automation and Mechatronics*, pp. 206–211, Singapore, June 2010.

[3] L. Tiseni, D. Chiaradia, M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli, "UV-C mobile robots with optimized path

planning: algorithm design and on-field measurements to improve surface disinfection against SARS-CoV-2," *IEEE Robotics & Automation Magazine*, vol. 28, no. 1, pp. 59–70, 2021.

[4] C. Xia and G. Xu, "The path planning algorithm studying about UAV attacks multiple moving targets based on voronoi diagram," *International Journal of Control & Automation*, vol. 9, no. 1, pp. 281–292, 2016.

[5] K. Zhang, P. Liu, W. Kong, Y. Lei, J. Zou, and M. Liu, "An improved heuristic algorithm for UCAV path planning," in *Proceedings of the Bio-Inspired Computing—Theories and Applications*, pp. 54–59, Xi'an, China, October 2016.

[6] G. Niu, Y. Zhang, and W. Li, "Path planning of continuum robot based on path fitting," *Journal of Control Science and Engineering*, vol. 2020, Article ID 8826749, 11 pages, 2020.

[7] J. Meng, V. M. Pawar, S. Kay, and A. Li, "UAV path planning system based on 3D informed RRT* for dynamic obstacle avoidance," in *Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1653–1659, Kuala Lumpur, Malaysia, December 2018.

[8] N. Geng, D. W. Gong, and Y. Zhang, "PSO-based robot path planning for multisurvivor rescue in limited survival time," *Mathematical Problems in Engineering*, vol. 2014, Article ID 187370, 10 pages, 2014.

[9] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[10] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[11] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[12] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2015.

[13] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[15] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[16] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," *Lecture Notes in Computer Science*, vol. 6, pp. 854–858, 2006.

[17] H. H. Inbarani, A. T. Azar, and G. Jothi, "Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 175–185, 2014.

[18] X. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.

[19] M. Zhu, S.-C. Chu, Q. Yang, W. Li, and J.-S. Pan, "Compact sine cosine algorithm with multigroup and multistrategy for dispatching system of public transit vehicles," *Journal of Advanced Transportation*, vol. 2021, Article ID 5526127, 16 pages, 2021.

[20] T. Joyce and J. M. Herrmann, "A review of no free lunch theorems, and their implications for metaheuristic optimisation," in *Nature-Inspired Algorithms and Applied Optimization*, X.-S. Yang, Ed., Springer International Publishing, Cham, Germany, pp. 27–51, 2018.

[21] Y. Zhang, G. Guan, and X. Pu, "The robot path planning based on improved artificial fish swarm algorithm," *Mathematical Problems in Engineering*, vol. 2016, Article ID 3297585, 11 pages, 2016.

[22] Y.-Q. Qin, D.-B. Sun, L. Ning, and Y.-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 4, pp. 2473–2478, Shanghai, China, August 2004.

[23] Q. Li, W. Zhang, Y. Yin, Z. Wang, and G. Liu, "An improved genetic algorithm of optimum path planning for mobile robots," in *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, vol. 2, pp. 637–642, Jian, China, October 2006.

[24] H.-J. Wang, Y. Fu, Z.-Q. Zhao, and Y.-J. Yue, "An improved ant colony algorithm of robot path planning for obstacle avoidance," *Journal of Robotics*, vol. 2019, Article ID 6097591, 8 pages, 2019.

[25] Q. Zhang, J. Ma, and Q. Liu, "Path planning based quadtree representation for mobile robot using hybrid-simulated annealing and ant colony optimization algorithm," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, pp. 2537–2542, Beijing, China, July 2012.

[26] H. Huang and C. Tsai, "Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm," in *Proceedings of the SICE Annual Conference 2011*, pp. 1338–1343, Tokyo, Japan, September 2011.

[27] T. Dao, T. Pan, and J. Pan, "A multi-objective optimal mobile robot path planning based on whale optimization algorithm," in *Proceedings of the 2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 337–342, Chengdu, China, November 2016.

[28] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.

[29] Y.-P. Xu, J.-W. Tan, D.-J. Zhu, P. Ouyang, and B. Taheri, "Model identification of the proton exchange membrane fuel cells by extreme learning machine and a developed version of arithmetic optimization algorithm," *Energy Report*, vol. 7, no. 12, pp. 2332–2342, 2021.

[30] P. Manoharan, P. Jangir, B. S. Kumar et al., "A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations," *IEEE Access*, vol. 9, pp. 84263–84295, 2021.

[31] Q. Yang, S.-C. Chu, J.-S. Pan, and C.-M. Chen, "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8184254, 10 pages, 2020.

[32] J. Abela and D. Abramson, *A Parallel Genetic Algorithm for Solving the School Timetabling Problem*, Division of Information Technology, C.S.I.R.O, Carlton, Australia, 1991.

[33] Q.-W. Chai, S.-C. Chu, J.-S. Pan, P. Hu, and W.-M. Zheng, "A parallel WOA with two communication strategies applied in DV-Hop localization method," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, p. 50, 2020.

[34] P.-W. Tsai, J.-S. Pan, S.-M. Chen, B.-Y. Liao, and S.-P. Hao, "Parallel cat swarm optimization," in *Proceedings of the 2008 International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3328–3333, Kunming, China, July 2008.

[35] M. S. Nasrabadi, Y. Sharafi, and M. Tayari, "A parallel grey wolf optimizer combined with opposition based learning," in *Proceedings of the 2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pp. 18–23, Bam, Iran, March 2016.

[36] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "Heuristic approaches in robot path planning: a survey," *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[37] M. N. Zafar and J. C. Mohanta, "Methodology for path planning and optimization of mobile robots: a review," *Procedia Computer Science*, vol. 133, pp. 141–152, 2018.

[38] P. Cui, W. Yan, and Y. Wang, "Reactive path planning approach for docking robots in unknown environment," *Journal of Advanced Transportation*, vol. 2017, Article ID 6716820, 11 pages, 2017.

[39] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, "A survey of parallel sequential pattern mining," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 3, p. 25, 2019.

[40] S. Pumma, M. Si, W. Feng, and P. Balaji, "Parallel I/O optimizations for scalable deep learning," in *Proceedings of the 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 720–729, Shenzhen, China, December 2017.

[41] J. F. Roddick, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, pp. 809–818, 2005.

[42] Y. Xin, Y. Liu, and L. Guangming, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[43] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: an effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Applied Soft Computing*, vol. 97, Article ID 106761, 2020.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3538462, 10 pages
https://doi.org/10.1155/2021/3538462

Research Article

The Application of Tree-Based Algorithms on Classifying Shunting Yard Departure Status

Niloofar Minbashi,[1] **Markus Bohlin**,[1] **Carl-William Palmqvist**,[2] **and Behzad Kordnejad**[1]

[1]*Division of Transport Planning, KTH Royal Institute of Techonology, 100 44 Stockholm, Sweden*
[2]*Division of Transport and Roads, Lund University, P.O. Box 118, 221 00 Lund, Sweden*

Correspondence should be addressed to Niloofar Minbashi; minbashi@kth.se

Received 16 April 2021; Accepted 13 August 2021; Published 8 September 2021

Academic Editor: Feng-Jang Hwang

Shunting yards are one of the main areas impacting the reliability of rail freight networks, and delayed departures from shunting yards can further also affect the punctuality of mixed-traffic networks. Methods for automatic detection of departures, which are likely to be delayed, can therefore contribute towards increasing the reliability and punctuality of both freight and passenger services. In this paper, we compare the performance of tree-based methods (decision trees and random forests), which have been highly successful in a wide range of generic applications, in classifying the status of (delayed, early, and on-time) departing trains from shunting yards, focusing on the delayed departures as the minority class. We use a total number of 6,243 train connections (representing over 21,000 individual wagon connections) for a one-month period from the Hallsberg yard in Sweden, which is the largest shunting yard in Scandinavia. Considering our dataset, our results show a slight difference between the application of decision trees and random forests in detecting delayed departures as the minority class. To remedy this, enhanced sampling for minority classes is applied by the synthetic minority oversampling technique (SMOTE) to improve detecting and assigning delayed departures. Applying SMOTE improved the sensitivity, precision, and F-measure of delayed departures by 20% for decision trees and by 30% for random forests. Overall, random forests show a relative better performance in detecting all three departure classes before and after applying SMOTE. Although the preliminary results presented in this paper are encouraging, future studies are needed to investigate the computational performance of tree-based algorithms using larger datasets and considering additional predictors.

1. Introduction

The "single wagonload" railway traffic has the potential to increase the modal share of rail freight transportation in Europe. The single wagonload traffic refers to wagonload shipments transported through a series of trains and shunting yards, instead of just on one train, from origin to destination [1]. In Europe, almost two-thirds of the single wagonload traffic is international; thus, promoting the single wagonload traffic can contribute to the economic growth of Europe by increasing international trades [2]. However, a recent study of 13 key countries of Europe showed that the single wagonload traffic shares only 27% of the total rail freight volume [2]. In fact, the single wagonload traffic loses a great part of its market to its road counterpart, particularly for small/medium shipments, due to low service reliability.

The low service reliability stems from the nature of single wagonload operations; wagons are detached from one train and attached to another train to continue their trip in typically large shunting yards. In European railways, shunting yard operations are high-priced from cost and time perspectives; in terms of costs, shunting and marshalling operations comprise 22% of the transport chain costs [2], and in terms of transit time, around 10–50% of the total transit time of freight trains is spent at shunting yards [3].

Increasing the predictability of shunting yard operations can improve the service reliability of single wagonload railway. The main outcome of shunting yard operations is

punctual train departures, and the predictability of train departures can be beneficial for both shunting yard operators and infrastructure managers. The former can use departure predictions to enhance shipment delivery times, whereas the latter can use it for improved planning of the interactions between the shunting yard departures and the punctuality of other trains on the line. Previously, shunting yards were considered as single entities to operate effectively without considering any interaction with the rest of the railway network. In recent years, however, the importance of analyzing the interaction between shunting yards and the railway network has increased [4, 5]. In fact, shunting yards are in constant interaction with the rest of the railway network; the lack of punctuality in receiving trains from the railway network can hinder train formations in shunting yards. On the contrary, the lack of punctuality in dispatching trains to the railway network may impact the punctuality of other trains in the railway network. In American railways, it was shown that the arrival time flexibility to shunting yards increases the average wagon dwell time and leads to wagons missing their departing train connections [5]. In European railways, a series of large-scale collaborative projects have been launched to model shunting yard-network interactions [4].

Predicting delayed departures from shunting yards, which is the focus of this paper, is clearly an important problem. In conjunction with the recent studies above, we therefore propose the application of tree-based machine learning algorithms to classify the status of departures from shunting yards. The approach presented in this paper is a preliminary step towards implementing an elaborate machine learning approach for departure delay prediction from shunting yards.

2. Related Work

The availability of large datasets in railways has led to the application of data-driven approaches for comprehensive railway operation analysis [6]. One of the methods to evaluate the quality of railway operations is combining the punctuality and delay measures [7], which has been studied extensively in three main areas: the prediction of train events, the prediction of train delays, and the propagation of train delays. In the prediction of train events, the main focus is on estimating running and dwelling times from predicting departure and arrival events [8, 9]. Delay prediction models aim at predicting primary delays of train arrivals and/or departures [10, 11]. Train delay propagation models express the development of secondary delays throughout a train journey by analyzing the impact of events, such as meetings and overtakings [12–14].

Since the scope of this paper is related to departure delay prediction models, a brief overview of the most relevant works is presented below. Previous research in this area has mainly focused on the arrival time estimation of passenger trains using data-driven approaches.

Wang and Work [15] proposed a historical regression model to predict passenger train delays before the beginning of the train trips in the US; the model was extended also to predict real-time delays using information from the previous stations and other trains on a corridor. Using data from Iranian railways, Yaghini et al. [16] showed that neural networks perform better than decision trees and multinomial logistic regression models in terms of training time and prediction accuracy. Marković et al. [10] were the first to apply the support vector regression for predicting passenger train arrival delays. Using data from Serbian railways, they concluded that results obtained from the support vector regression performed better than artificial neural networks. Oneto et al. [17–20] provided an extensive study of big data analytics implemented in a train delay prediction system for large-scale railway networks with data from Italian railways. In their papers, they proposed the application of shallow and deep extreme learning machines for trains' delays. Nair et al. [21] developed a large-scale ensemble passenger train delay model in German railways. By combining a statistical random forest-based model, a kernel regression model, and a mesoscopic simulation model, they demonstrated a 25% improvement potential in the prediction accuracy and 50% reduction in root mean squared errors compared to the published schedule.

Although the number of studies using data-driven approaches for passenger train delays is substantial, the application of these approaches for the freight train delay prediction is quite recent. The main reason is that passenger and freight trains differ inherently in stopping patterns, dispatching priority, and train characteristics. In general, implementing delay prediction models for freight trains in mixed railway networks is more complex due to the prioritized running of passenger trains [22]. This prioritization sometimes imposes initial departure delays and/or long meeting and overtaking times during freight train runs, which may lead to delayed arrivals as well, all of which are complicated to mathematically grasp in a delay prediction model for freight trains. Apart from this, freight train operators may not be willing to share the operational data, as it may contain commercially sensitive information.

In a freight train delay context, Gorman [23] applied econometric methods for predicting congestion delays using data from the US. Barbour et al. [24] implemented a support vector regression model for estimating train arrival (ETAs) of individual freight trains. They achieved an improvement of 21% over a baseline prediction method at some locations and average 14% across the study area. Later, Barbour et al. [11] compared the predictive performance of linear and nonlinear support vector regression, random forest regression, and deep neural network models using data from the US. They showed that the maximum ETA error reduction of support vector machines and deep neural networks was 26% better than a statistical baseline predictor. They achieved the best performance in the random forest models, which achieved an error reduction of 60% compared to the baseline predictor at some points, and an average error reduction of 42%.

So far, studies on freight train delays have been dedicated to arrival delays, whereas the nature of departure and arrival delays differ for freight trains [25]. Arrival delays are the result of the accumulation of delays along the train journey, whereas departure delays are the result of shunting yard

improper functioning. In Sweden, delayed departures from shunting yards were one of the five main causes of delays due to operator error [26].

3. Method

In this section, the data, the specifics of the shunting yard departure status prediction problem, the application of supervised machine learning, and the two applied machine learning methods (decision trees and random forests) are presented.

3.1. Data. In this paper, we combine two different datasets from the Hallsberg yard in Sweden, the largest shunting yard in Scandinavia, provided by the main yard operator in Sweden. Hallsberg has the conventional European shunting yard characteristics [27] and layout, which comprises of three subyards for arrivals, classification, and departures, respectively (see Figure 1). Trains are received in the arrival yard, where their wagons are then decoupled. Then, via a hump (a small hill), the wagons are rolled to the classification yard, where the arrangement of wagons is changed to form new trains for new destinations. When departing trains are ready, they are sent to the departing yard, where the locomotive is attached and the train is prepared to be dispatched to the network.

The first dataset used in this paper is a wagon connection dataset, which gives the information about connections between arriving and departing trains, in particular, information on which wagon arrived in which train and departed by which train. The total number of connections between an arriving train and a departing train were 6,243 (representing 21,381 wagon connections) in a one-month period, October 2015. The second dataset contains train punctuality data giving the actual arrival and departure times of trains in minutes.

As the obtained datasets did not cover a large number of parameters, the predictors that could be extracted from these two datasets for modelling were limited, but chosen to represent the performance of the three subyards to a reasonable extent. Table 1 shows the selected predictors for each subyard.

In Swedish practice, freight train departures are typically classified as early, on-time, or delayed. In particular, any train departing before their scheduled departure time is classified as early, and shunting yard operators are allowed to dispatch trains early provided that there is free capacity slot on the line. This is a common practice in Swedish railways to compensate for further disturbances that might occur during a freight train run. Trains that depart with a delay of at most five minutes from their scheduled departure time are furthermore classified as on-time. Any deviation over five minutes from the scheduled departure time indicates a delayed departure.

3.2. Problem Definition. Shunting yard departure prediction is, in essence, a complex problem due to the following three aspects:

(i) Departure deviations typically have long tails, and depicting the histogram of departure deviations shows the long tails in both positive and negative values (see Figure 2). It is already difficult to fit a probability distribution to such deviations, and they are notoriously difficult to model in machine learning. Because of this, performance generally suffers when the data exhibit this characteristic.

(ii) The departure status classes show great disparity. For example, as shown in Figure 3, the majority of departures in the considered dataset are early (65%), whereas the share of on-time (19%) and delayed departures is almost similar (15%). This disparity makes the dataset imbalanced and makes predictions biased towards the majority class. This is discussed further in Section 3.5.

(iii) Shunting yard operations are highly human-dependent, and almost all disturbances in shunting yard operations are handled by the shunting personnel. However, most of these human interventions are not discernible in the dataset. This makes the modelling process difficult since we cannot allocate proper model parameters to these human interventions. Furthermore, in later stages of the modelling, this potential source of error makes a proper interpretation of any predictions much harder to do.

In the big picture, shunting yard departure prediction can be decomposed into different levels with different priorities from the shunting yard operator and infrastructure manager perspectives. The first level is to classify the departure status; delayed departures comprise a small part of the dataset, but are critical for both the shunting yard operator and the infrastructure manager. In addition, delayed departures have a different distribution from early departures [25, 28], which may result in distinct models for delayed departures. Once delayed departures are classified, the actual delay can be predicted in the second level [21]. In the third level, delayed departures can be mitigated by rebooking delayed wagons to different trains in order to minimize the departure delays. This paper considers only modelling and evaluation of the first level, i.e., classification of departure status.

3.3. Supervised Machine Learning for Shunting Yard Departure Status Prediction. We applied the concept of supervised learning to predict shunting yard departure status using the KNIME analytics platform [29]. For our specific problem, the supervised learning can be described as follows. We implement a machine learning model by taking a set of training data on departing trains' parameters with known departure status. Then, we minimize a prediction error for predicting the output which classifies the departure status into delayed, early, or on-time classes using another part of the data called test data; this process is depicted schematically in Figure 4.

Figure 1: A conventional European shunting yard layout [28].

Table 1: The selected predictors.

Subyard	Predictors
Arrival	Scheduled arrival time Scheduled arrival date Actual arrival time Actual arrival date Arriving train number Train arrival time deviation
Classification	Wagon standing time
Departure	Scheduled departure time Scheduled departure date Actual departure time Actual departure date Departing train number

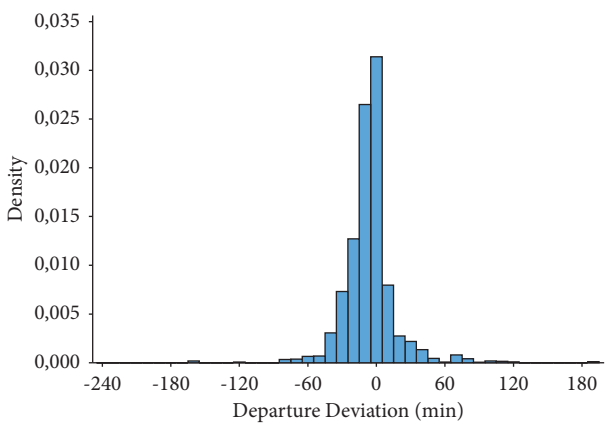


Figure 2: Histogram of departure deviations.

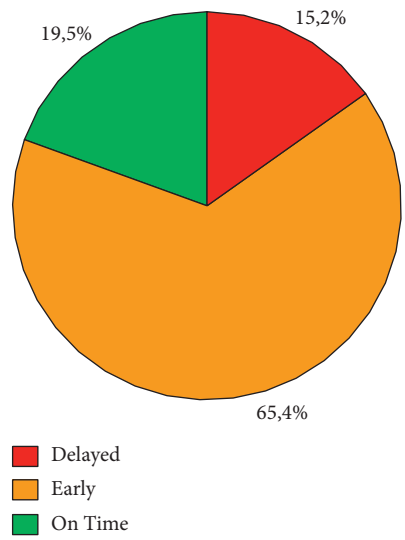


Figure 3: Pie chart of departure status.

3.3.1. Decision Tree. Decision trees are based on a hierarchy of if/else questions resulting in a decision. A tree is made up of nodes and branches; each node represents either a question about an input feature or an end node (a leaf which is associated to a class). At each node, the data is branched on an input feature generating two or more branches. As the tree develops, more branches partition the original data; this procedure is continued until no further branches are possible. The final goal of a decision tree algorithm is to partition the training dataset into subsets until each partition is either "pure" (containing only samples of one class) in terms of the target class or sufficiently small.

3.3.2. Random Forest. Random forest is an ensemble version of decision trees; ensemble methods compound different machine learning models to overcome the low performance of each model and create a robust model. Therefore, a random forest is a collection of decision trees, where each decision tree is trained on a subset of the original dataset. Each tree might perform well on a subset of the dataset, but it might overfit on a part of the subset. Therefore, combining various trees which perform well but overfit in different parts of the data can decrease the overfitting by averaging the results, while maintaining the predictive ability of each tree.

3.4. Resampling Procedure and Splitting Criteria. Resampling techniques are used to estimate model performance; by resampling, the model is trained by a subset of data and the rest of data is used to evaluate the model efficiency. The method in which the subsets are resampled is important to overcome bias and variance of the model generalization. We applied a common 10-fold cross-validation for the resampling procedure [31]. After sampling, we specify how to partition the data to reach the purest subset, where a pure group contains a larger proportion of one class in each node. Purity can be achieved by maximizing accuracy or minimizing misclassification error. However, in maximizing accuracy, the focus is on partitioning the data with minimal misclassification, whereas in maximizing purity, we mainly aim at partitioning the data by placing the samples in one particular class. The Gini index [32] is a measure that focuses on purity and is often used for this purpose. Basically, at each partitioning step, the split point

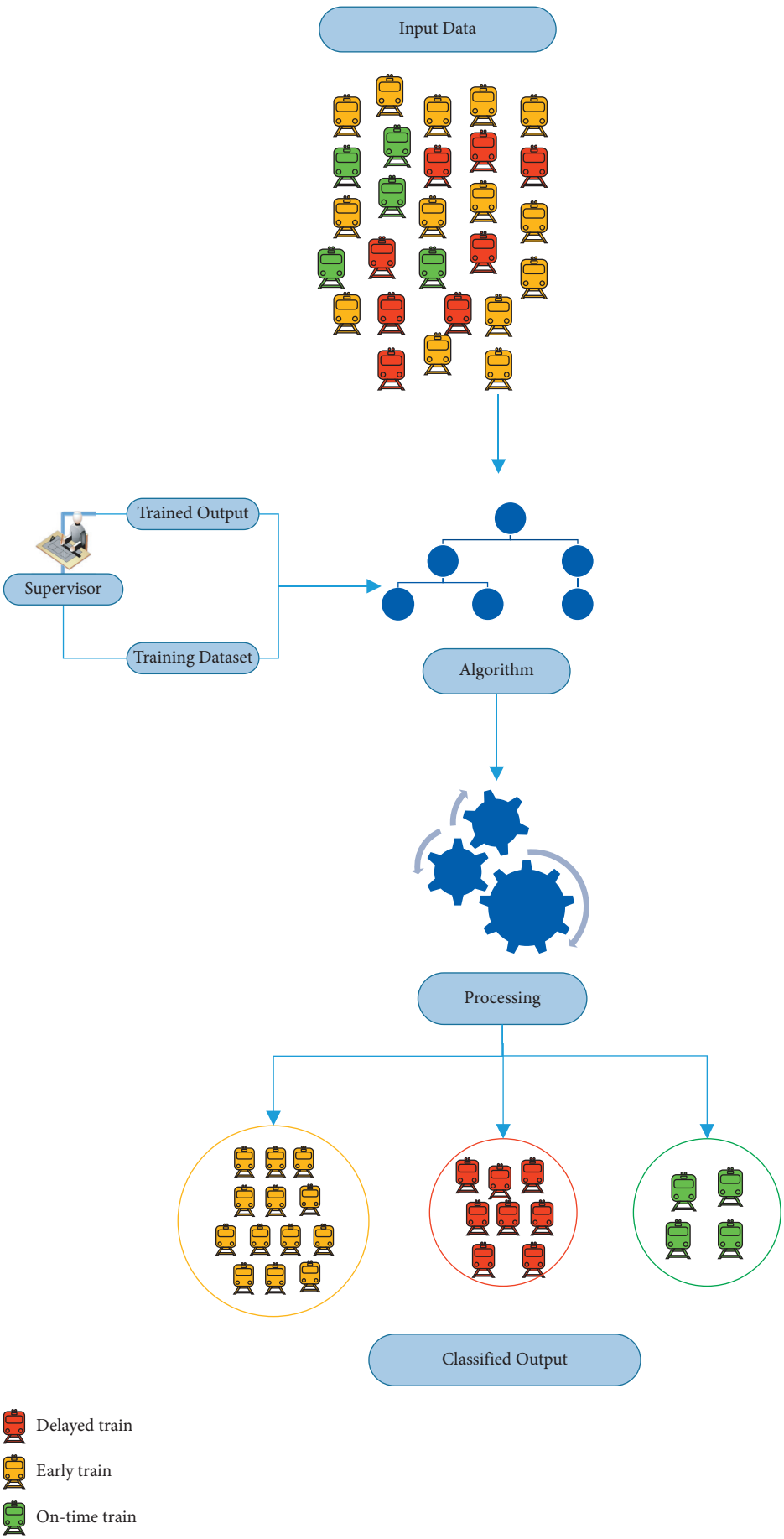


Figure 4: Method (modified from [30]).

value that minimizes the impurity is selected. In this paper, we use the formulation in [31] for calculating the Gini index.

3.5. Decision Tree and Random Forest Improved by SMOTE. In many classification problems, some classes are typically more important for the modeller than others, but the number of instances of those classes in the dataset is in minority. Examples from the railway domain include classification of maintenance needs for vehicles or track and signaling equipment. When using machine learning, this situation may lead to imbalanced learning, when a majority of the instances are erroneously predicted to belong to the majority class. Imbalanced learning can be combated through three different approaches: problem redefinition, data-level approaches, or algorithm-level approaches [33]. In our dataset, the majority of the instances are early trains. To examine the model performance considering the imbalanced dataset, we applied the data-level sampling technique called synthetic minority oversampling technique (SMOTE) for training the model. SMOTE oversamples the data from the minority classes by averaging on a number of the nearest neighbours [33]. In this paper, we used 15 nearest neighbors and oversampled the delayed and on-time minority classes.

3.6. The Evaluation Criteria. There are various criteria for evaluating the performance of a machine learning model. The selection of these criteria depends on the purpose of the model and the modeller's choice. In our model, the primary interest is to have an acceptable performance on the delayed class, and the secondary interest is to evaluate how good the model performance is on classifying all three classes. Therefore, we used the confusion matrix to compare the results of each model. The confusion matrix summarizes whether the observed instances are correctly or incorrectly classified. In general, the table cells indicate the number of the true positives (TP: correctly classified in the positive class), false positives (FP: incorrectly classified in the positive class), true negatives (TN: correctly classified in the negative class), and false negatives (FN: incorrectly classified in the negative class). Table 2 shows a schematic view of the confusion matrix to classify the delayed class. Here, the delayed class is the positive class and the two other classes are the negative ones. The same procedure is conducted for the two other classes.

To evaluate how the model performs on a single-class level, we calculated precision, sensitivity, specificity, and F-measure. To evaluate the overall performance on a multiclass level, we calculated overall accuracy and Cohen's Kappa [31].

Sensitivity is calculated in equation (1), and it evaluates how good the model is in detecting positive instances.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{1}$$

Precision, in equation (2), evaluates how precise the model is when assigning instances of a given class. More precisely, it evaluates the proportion of instances assigned to a positive class that are truly positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

Specificity, calculated in equation (3), is the compliment measure to the sensitivity and evaluates how good the model is in detecting the negative instances.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{3}$$

Sensitivity and precision are the most important criteria in our model assessment; F-measure in equation (4) is the harmonic mean of sensitivity and precision, and it balances the use of sensitivity and precision in the model evaluation.

$$F - \text{measure} = 2 \cdot \frac{\text{sensitivity} \cdot \text{precision}}{\text{sensitivity} + \text{precision}}. \tag{4}$$

Overall accuracy is calculated by dividing the total number of correctly classified instances to the total number of instances. Overall accuracy does not make any distinction between classes and evaluates overall results.

The last criterion is Cohen's Kappa which shows the level of agreement between the predicted classes and the actual ones. Particularly, it is a good measure for evaluating model performance when the classes are imbalanced. Cohen's Kappa is calculated in equation (5), where P_o and P_e represent, respectively, the overall accuracy of the model and the measure of the agreement between the model predictions and the actual class values as if happening by chance.

$$\text{Cohen's Kappa} = \frac{P_o - P_e}{1 - P_e}, \tag{5}$$

$$P_e = \sum_{i=1}^{3} P_{ei,\text{actual}} \cdot P_{ei,\text{predicted}}. \tag{6}$$

The model predictions and actual class values are assumed to be independent, and P_e sums P_e of all classes in equation (6), where $P_{ei,\text{actual}}$ is the proportion of actual class values (TP + FN) from the total number of instances and $P_{ei,\text{predicted}}$ is the proportion of predicted class values (TP + FP) from the total number of instances [34]. In equation (5), $P_o - P_e$ shows the difference between the observed overall accuracy of the model and the overall accuracy that can be obtained by chance, and $1 - P_e$ stands for the maximum value for this difference. Cohen's Kappa lies between 0 and 1. When the observed difference and the maximum difference are close to each other, Cohen's Kappa is close to 1 representing a good model, whereas in a random model, the overall accuracy depends on the random chance, so the difference $P_o - P_e$ is zero, and Cohen's Kappa is 0.

Overall, when Cohen's Kappa is closer to 0, the agreement between the actual classes and predicted classes is lower, whereas the closer it is to 1, the agreement between the actual classes and predicted classes is higher. In general, Cohen's Kappa can be interpreted, as shown in Table 3 [35].

Table 2: The sample confusion matrix for positive class (delayed) and negative class (early and on-time).

Observed	Classified		
	Delayed	Early	On-time
Delayed	TP	FN	FN
Early	FP	TN	TN
On-time	FP	TN	TN

Table 3: Interpretation of Cohen's Kappa [35].

κ	Interpretation
0	Agreement equivalent to chance
0.10 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 0.99	Near-perfect agreement
1	Perfect agreement

4. Results and Discussion

We implemented the models in the KNIME analytics platform [29]. Table 4 compares results from all four models. First, we discuss the results for the delayed class since it is the primary interest of this paper. Then, we discuss the generality of the model in classifying all three classes.

Delayed departures comprise a small portion of the departures, which makes their predictability difficult; comparing the decision tree and random forest models before using SMOTE shows this difficulty in terms of sensitivity and precision parameters. The models detect approximately one third of the delayed departures (33% for the decision tree model and 28% for the random forest model). The models are further not precise enough to assign the delayed departures to the delayed class; precision of the two models are slightly better than the sensitivity: 39% and 49% for decision tree and random forest, respectively.

In our model evaluation, both sensitivity and precision are important parameters to evaluate the model in detecting and assigning the positive instances. However, F-measure, the combination of these two parameters, shows similar low performance of the two models: decision tree (35%) and random forest (36%). In problems with imbalanced data, F-measure is an appropriate measure to compare the models. However, when we examine F-measure to compare the decision tree and random forest, in terms of the generalization for all three classes, we see the disparity between the F-measures for the three classes. The imbalanced dataset makes it difficult to select between the two models; the F-measure is high for the majority class 76% in decision tree and 82% in random forest and very low for minority classes.

The imbalanced proportion of the class instances reflects the low performance of the decision tree and random forest, especially in the delayed class. Therefore, the results are also compared after having imbalanced data modified by SMOTE. For the delayed class, in the decision tree model, an improvement of 18% in sensitivity and 20% in precision is observed. The improvement for random forest in these two parameters is 29% and 28%, respectively. The F-measure is also improved with almost the same proportion, 20% for decision trees with SMOTE and 26% for the random forest with SMOTE. Using SMOTE mainly balanced the F-measure in the three classes in both models, which shows that models are improved in terms of overall classification of the three classes after having applied SMOTE. The major improvement that SMOTE made to both models was balancing the specificity in all three classes which means that the models assign the same proportion of false positives to the three classes. However, it is observed that specificity and sensitivity for all three classes are inversely related, which is often the case in classification problems with imbalanced classes. In general, selecting the optimal balance between sensitivity and specificity in a classification model is dependent on the goal of the classification purpose. In this paper, the main interest was on the minority class of delayed departures.

Considering the small initial dataset of one month and the specific purpose of this paper, we concluded two main conjectures: first, using tree-based algorithms to classify departures without treating the imbalanced data may not give a satisfactory level of sensitivity and precision, especially for the minority class. In addition, random forest as an ensemble method showed better results than a single decision tree model. The results from an ensemble model are preferred as they reduce the bias and variance. This is also reflected in the measure of Cohen's Kappa for the random forest model with SMOTE at 0.53, which shows moderate agreement between the predicted classes and the actual classes. It is worthy to note that overall accuracy does not show much improvement after applying SMOTE in models since overall accuracy does not reflect the imbalanced classes. Thus, it may not be an adequate criteria in our model assessment.

The final point of reflection is the predictors used for classification. We used a small set of predictors that were present in our dataset. These predictors may not represent departures entirely. There are other predictors that may have more impact on the departures, such as weather condition parameters, train characteristics, and the experience level of shunting yard operator staff. One of the limitations in studying shunting yards is the complexity of obtaining these data; shunting yards have security importance for infrastructure managers; in addition, shunting yard operators may not be willing to share most of their data due to business-related issues.

Table 4: Class statistics for all models.

Model	Class	True positives	False positives	True negatives	False negatives	Sensitivity (%)	Precision (%)	Specificity (%)	*F*-measure (%)	Cohen's Kappa	Overall accuracy (%)
Decision tree	Delayed	308	486	4778	633	33	39	91	35	0.28	64
	Early	3195	1097	1051	862	79	74	49	76		
	OnTime	499	620	4378	708	41	45	88	43		
Decision tree (SMOTE)	Delayed	2084	1450	6664	1973	51	59	82	55	0.42	61
	Early	3090	1516	6598	967	76	67	81	71		
	OnTime	2297	1734	6380	1760	57	57	78	56		
Random forest	Delayed	262	267	4997	679	28	49	95	36	0.39	72
	Early	3644	1162	986	413	90	76	46	82		
	OnTime	592	278	4720	615	49	68	94	57		
Random forest (SMOTE)	Delayed	2323	1158	6956	1734	57	67	86	62	0.53	68
	Early	3419	1425	6689	638	84	71	82	77		
	OnTime	2580	1266	6848	1477	64	67	84	65		

5. Conclusion

n this paper, we discussed the problem of classifying departure status from shunting yards as a first step to implement an elaborate departure prediction model for shunting yards in the future. Departure status from shunting yards impact the punctuality of other trains on the network. A delayed departure from the shunting yard may delay consequent freight train departures or interfere with passenger trains running on the line connected to the shunting yard. Additionally, delayed departures from shunting yards can mostly lead to delayed arrivals to the next shunting yard causing delayed shipment delivery and customer loss for rail freight operators. One of the main advantages of shunting yard departure prediction models is assisting infrastructure managers in improved allocation of capacity on the lines connected to the shunting yards. Particularly, in a railway context such as Swedish railways, where the infrastructure manager and yard operator are two different stakeholders, the infrastructure manager requires more transparency from the yard operation side to control its impact on the punctuality of the network. On the contrary, these models help shunting yard operators to become more agile in better utilization of train capacity when replanning of wagons is required due to missed connections. However, there is a lack of practical models for shunting yard departure prediction in the previous literature; to the best of our knowledge, no previous research has been conducted to predict the status of departures from shunting yards.

We aimed at comparing the performance of tree-based (decision tree and random forest) algorithms which have shown an overall adequate performance in comparison with other machine learning algorithms on delay prediction in the previous research. Typically, the departure status from shunting yards are imbalanced; delayed departures are in minority, whereas they are arguably much more important to predict. Our results showed that decision trees and random forests both require oversampling to improve the sensitivity and precision in classifying delayed departures. Applying the synthetic minority oversampling technique (SMOTE) in both models improved the sensitivity, precision, and *F*-measure of delayed departures. The improvement was approximately 20% for decision tree and 30% for random forest. We achieved the overall best results using the random forest algorithm improved by SMOTE. By applying SMOTE to our small case study, we showed how promising tree-based algorithms can be for departure status prediction when larger datasets are available.

Hence, we propose future directions that can contribute to improved insight into shunting yard departure status prediction. First, it would be beneficial to examine the computational performance of decision tree and random forest algorithms using larger datasets. Second, there are other predictors that affect shunting yard departures; adding operational parameters, train characteristics parameters, weather condition parameters, and parameters representing shunting yard staff performance may enhance the accuracy of the models. Third, future attempts can use shunting yard departure status models to develop models for predicting the actual departure time deviations. Such models can then be combined with network simulation models to analyze the overall shunting yard-network interactions.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Disclosure

Some of the results presented in this paper have previously been presented in the final deliverable of the European FR8HUB project [30].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Magnus Wahlborg and Fredrik Lundström from the Swedish Transport Administration (Trafikverket) for their continuous support during the time of conducting this research and Jonatan Gjerdrum from Green Cargo for providing the data and sharing his expertise to improve our analysis. This research was

supported by Trafikverket (the Swedish Transport Administration) (Grant no. TRV2017/68055) and FR8HUB Project within the European H2020-Shift2Rail IP5 (Grant agreement no. 777402—FR8HUB—H2020-S2RJU-2017/H2020–S2RJU-CFM-2017).

References

[1] J. Dirnberger, "*Development and application of lean railroading to improve classification terminal performance*," M.S. thesis, University of Illinois at Urbana-Champaign, Springfield, IL, USA, 2006.

[2] P. Guglielminetti, M. F. Lagraulet, D. Artuso et al., "Study on single wagonload traffic in europe-challenges, prospects and policy options final report," Technical Report, Sapienza University of Rome, Rome, Italy, 2015.

[3] Y. Bontekoning and H. Priemus, "Breakthrough innovations in intermodal freight transport," *Transportation Planning and Technology*, vol. 27, no. 5, pp. 335–345, 2004, https://www.tandfonline.com/action/journalInformation?journalCode=gtpt20.

[4] S2R Joint Undertaking, "SHIFT2RAIL strategic master plan," Technical Report, The European Commision, Brussels, Belgium, 2015.

[5] C. T. Dick and N. Nishio, "Influence of mainline schedule flexibility and volume variability on railway classification yard performance," in *Proceedings of the RailNorrköping 2019. 8th International Conference on Railway Operations Modelling and Analysis (ICROMA)*, A. Peterson, M. Joborn, and M. Bohlin, Eds., pp. 406–425, Linköping, Sweden, July 2019.

[6] F. Cerreto, B. F. Nielsen, O. A. Nielsen, and S. S. Harrod, "Application of data clustering to railway delay pattern recognition," *Journal of Advanced Transportation*, vol. 2018, Article ID 6164534, 9 pages, 2018.

[7] G. Medeossi and S. De Fabris, "Simulation of rail operations," *Handbook of Optimization in the Railway Industry*, vol. 268, pp. 1–24, 2018.

[8] P. Kecman and R. M. Goverde, "Online data-driven adaptive prediction of train event times," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 465–474, 2015.

[9] P. Kecman and R. M. P. Goverde, "Predictive modelling of running and dwell times in railway traffic," *Public Transport*, vol. 7, no. 3, pp. 295–319, 2015.

[10] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 251–262, 2015.

[11] W. Barbour, C. Samal, S. Kuppa, A. Dubey, and D. B. Work, "On the data-driven prediction of arrival times for freight trains on u.s. railroads," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2289–2296, IEEE, Maui, HI, USA, 2018.

[12] J. Yuan, *Stochastic modelling of train delays and delay propagation in stations*, Ph.D. dissertation, TU Delft, Delft, Netherlands, 2006.

[13] W.-H. Lee, L.-H. Yen, and C.-M. Chou, "A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 49–64, 2016.

[14] S. Harrod, F. Cerreto, and O. A. Nielsen, "A closed form railway line delay propagation model," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 189–209, 2019.

[15] R. Wang and D. B. Work, "Data driven approaches for passenger train delay estimation,," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 535–540, Institute of Electrical and Electronics Engineers Inc., Gran Canaria, Spain, October 2015.

[16] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *Journal of Advanced Transportation*, vol. 47, no. 3, pp. 355–368, 2013, http://doi.wiley.com/10.1002/atr.193.

[17] L. Oneto, E. Fumeo, G. Clerico et al., "Advanced analytics for train delay prediction systems by including exogenous weather data," in *Proceedings-3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 458–467, Montreal, QC, Canada, October 2016.

[18] L. Oneto, E. Fumeo, G. Clerico et al., "Advanced analytics for train delay prediction systems by including exogenous weather data," in *Proceedings-3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 458–467, Institute of Electrical and Electronics Engineers Inc., Montreal, QC, Canada, October 2016.

[19] L. Oneto, E. Fumeo, G. Clerico et al., "Dynamic delay predictions for large-scale railway networks: deep and shallow extreme learning machines tuned via thresholdout," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2754–2767, 2017, http://ieeexplore.ieee.org/document/7917288/%20.

[20] L. Oneto, E. Fumeo, G. Clerico et al., "Train delay prediction systems: a big data analytics perspective," *Big Data Research*, vol. 11, pp. 54–64, 2018.

[21] R. Nair, T. L. Hoang, M. Laumanns et al., "An ensemble prediction model for train delays," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 196–209, 2019.

[22] M. H. Dingler, Y.-C. Lai, and C. P. Barkan, "Impact of operational practices on rail line capacity: a simulation analysis," in *Proceedings of the 2009 Annual AREMA Conference*, Chicago, IL, USA, June 2009.

[23] M. F. Gorman, "Statistical estimation of railroad congestion delay," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 3, pp. 446–456, 2009.

[24] W. Barbour, J. C. Martinez Mori, S. Kuppa, and D. B. Work, "Prediction of arrival times of freight traffic on US railroads using support vector regression," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 211–227, 2018.

[25] N. Minbashi, C.-W. Palmqvist, M. Bohlin, and B. Kordnejad, "Statistical analysis of departure deviations from shunting yards: case study from Swedish railways," *Journal of Rail Transport Planning & Management*, vol. 18, p. 6, 2021, https://linkinghub.elsevier.com/retrieve/pii/S2210970621000159.

[26] N. A. Krüger, I. Vierth, and F. F. Roudsari, "Spatial, temporal and size distribution of freight train delays: evidence from sweden," 2013, http://www.diva-portal.org/smash/get/diva2:1157840/FULLTEXT01.pdf.

[27] Shift2Rail joint undertaking, "smart automation of rail transport (SMART): deliverable D5.2 integration data in unique database of EU marshalling yards," Technical Report, TU Delft, Delft, Netherlands, 2017.

[28] N. Minbashi, *Applying Data Analytics to Freight Train Delays in Shunting Yards*, KTH Royal Institute of Technology, Stockholm, Sweden, 2020, http://www.diva-portal.org/smash/record.jsf?pid=diva2:1485378.

[29] "KNIME Analytics Platform." [Online]. Available: https://www.knime.com/knime-analytics-platform.

[30] M. Kuhn and K. Johnson, "Applied predictive modeling," Springer, New York, NY, USA, 2013.

[31] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees," *Wadsworth Int," Group*, vol. 37, no. 15, pp. 237–251, 1984.
[32] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2013.
[33] M. Widmann and A. Roccato, *From Modeling to Model Evaluation*, KNIME Press, Zurich, Switzerland, 2021.
[34] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.
[35] Shift 2 Rail Joint Undertaking, "FR8HUB: deliverable 3.3 results of traffic simulation of defined scenarios and evaluation," Technical Report, TU Delft, Delft, Netherlands, 2020.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9426659, 8 pages
https://doi.org/10.1155/2021/9426659

Research Article

Optimization of Distribution Path considering Cost and Customer Satisfaction under New Retail Modes

Dengqing Wang,[1,2] **Yuting Yang**,[2] **and Yanhu Wang**[3]

[1]*School of Economics and Management, Fuzhou University, Fuzhou 350108, China*
[2]*School of Economics and Management, Fujian Chuanzheng Communications College, Fuzhou 350007, China*
[3]*The Academy of Digital China (Fujian), Fuzhou University, Fuzhou 350108, China*

Correspondence should be addressed to Dengqing Wang; 178477427@qq.com

Received 21 May 2021; Revised 24 July 2021; Accepted 17 August 2021; Published 7 September 2021

Academic Editor: Yuan Gao

One of the top issues in logistics management and related research is to establish an effective distribution system that is adaptive to new retail and capable of lowering the cost of logistics while enhancing consumer satisfaction. Aimed at reversing the weak points of current logistics distribution patterns, a dual-objective bipolar model with optimal logistics cost and consumer satisfaction by restraining distribution time and load is tested in this paper to figure out the proper nodes and vehicle routes. Data from general and front warehouses of PuPu mall, a Fuzhou-based online retail enterprise, are made into a case study. Moreover, the immune algorithm and genetic algorithm are adopted to achieve the model solution. It is found that the immune algorithm is more efficient than the genetic algorithm in searching solutions, thus having better adaptivity and effectiveness, and also that the type of distribution vehicle plays a significant role in determining the total distribution cost.

1. Introduction

In recent years, online retail (also known as new retail) has shown a rapid growth trend. In 2020, China's online retail sales reached 11760.1 billion yuan, an increase of 10.9% over the previous year. Among them, the online retail sales of physical goods reached 9759 billion yuan with an increase of 14.8%, accounting for 24.9% of the total retail sales of social consumer goods. Logistics generation and order generation are the two basic links of online retail, and the evolution of the combination mode of the two constantly reshapes the format of e-commerce, resulting in the rapid updating and iteration of new retail modes. At present, the mainstream of the industry consists of three trends: integrating offline with online (such as https://www.Suning.com), integrating online with offline plus self-run logistics service (such as https://www.JD.com), and online order plus front warehouse distribution and delivery (such as https://www.pupumall.com, Fuzhou). The distribution networks of these new retail logistics have the following common characteristics: they directly face the terminal customers, they have adjacent distribution outlets for customers and fast circulation, there are high requirements for customized and timely delivery, and there is low certainty of distribution quantity and frequency. A key problem of the new retail industry to solve as well as a research hotspot in related fields is to figure out how to adapt to the changes of new retail logistics distribution to reduce logistics costs and improve customer satisfaction, the core of which is to optimize the logistics path, including the optimization of logistics nodes and the paths from terminal nodes to distribution terminals (addresses of online shopping customers).

Scholars at home and abroad have taken large-scale online-to-offline (O2O) retail enterprises (such as Suning Appliance), their offline experience stores and distribution networks, and their terminal customers (mainly in cities where e-commerce is well developed) as research objects to reach various research targets [1, 2], such as minimizing total logistics cost, minimizing driving distance [3, 4], achieving greater customer satisfaction with the optimal distribution

cost, or having the best goods circulation ability and logistics service level. The proposed models were solved by various algorithms to establish the optimal logistics distribution network nodes and paths [5–8]. Generally, the genetic algorithm [9–12] and immune algorithm [13–15] are widely applied to solve practical models. For example, Hou et al. [10] tried to develop a given refining schedule for the crude oil operations, which were transformed into an assignment problem of charging tanks and distillers. The problem was solved by a genetic algorithm. Wang et al. [11] constructed a multiperiod portfolio model by considering a conditional value-at-risk constraint for the multiperiod investment decisions. The model was solved by a variable neighborhood search-based hybrid genetic algorithm. Guo et al. [12] considered a multiobjective resource-constrained and sequence-dependent disassembly optimization problem with disassembly precedence constraints. This multiobjective optimization problem with a combinatorial nature was solved by a genetic algorithm efficiently.

However, some parts have still been left behind and need to be further tapped into the literature. First, there has not been pertinent and comprehensive research on the logistics distribution networks of regional new retail enterprises in China. Second, there is no comprehensive analysis of the influencing factors of customer satisfaction and minimizing the logistics cost of outlets has not been involved in the overall consideration. Third, there is no in-depth analysis of the impact of the influencing factors on customers' choice of front warehouses on logistics network operating costs.

In view of the limitations of the existing research results on new retail logistics terminal node layout and considering customer satisfaction and total logistics cost, this paper constructs a dual-objective bilevel programming model for site selection and layout optimization of new retail distribution centers and their front warehouses. An immune algorithm is adopted to seek solutions, aiming to provide technical support for solving new retail logistics problems.

2. Problem Prototypes and Models

2.1. Problem Description. Based on the logistics distribution network of https://www.pupumall.com, a regional new retail enterprise, this paper investigates the solutions to this kind of network by modeling, which involves optimizing distribution lines from the general warehouse to each front warehouse, selecting the sites of front warehouses, and covering the service area. The general warehouse is responsible for storage, sorting, transportation, distribution, distribution processing, order information processing, and so on, as well as delivering goods to the front warehouses within the region. The front warehouses are located where there is easy and quick access to customers and are responsible for temporary storage, transfer, sorting, order processing, and other functions, and it is required that goods be delivered within 30 minutes after receiving a customer's order. Light trucks are used for transportation from the general warehouse to the front warehouses, while electric vehicles are used from the front warehouses to the terminal customers. The total cost of logistics distribution includes both the transportation cost from the general warehouse to the front warehouses and the distribution cost from front warehouses to customers.

Customer satisfaction with the front warehouses is determined by factors such as distribution time, pickup distance, service price, service quality, and goods turnover ability [7]. As https://www.pupumall.com delivers goods directly to the door without the need for customers to pick up the goods themselves, customer satisfaction mainly depends on the distribution time, since the logistics distribution market is quite transparent and the prices are consistent, so they do not need to be considered. Therefore, customer satisfaction with the front warehouses is mainly influenced by three factors: distribution time, turnover capacity, and service quality.

2.2. Solution Model. To facilitate the search for a solution and increase the practical value of the model, the following assumptions are made for the logistics networks in this study:

(i) There is only one general warehouse, which is responsible for supplying goods to each front warehouse

(ii) The maximum processing capacity of an alternative front warehouse is the same as that of those in use, and the location is known

(iii) Electric vehicles are employed to transport goods from front warehouses to terminal customers and are capable of circular distribution

(iv) The maximum capacity of each vehicle is limited

(v) The demand for goods from each front warehouse is met, and only one vehicle services the goods

(vi) The length of each distribution path shall not exceed the maximum driving distance of the vehicle in service

(vi) The fixed cost of each front warehouse is different

(vii) The delivery time from a front warehouse to a customer is expected to be within 30 minutes

Moreover, the model sets, parameters, and decision variables used throughout the paper are as follows:

Sets:

K is the set of vehicles used between the general warehouse and front warehouses
N is the set of vehicles used between the general warehouse and front warehouses
I is the collection of front warehouses
J is the collection of locations of terminal customers
N is the collection of terminal distribution of electric vehicles

Index:

o is the index of general warehouse
k is the index of vehicles of vehicle set K
n is the index of vehicles of vehicle set N
i is the index of front warehouses
j is the index of terminal customers

Parameters:

c_1 is the transportation rate from the general warehouse to a front warehouse (RMB/(kg·km))
c_2 is the transportation rate from a front warehouse to a terminal customer (RMB/(kg·km))
k_1 is the weight of the customer satisfaction on distribution time
k_2 is the weight of the customer satisfaction on turnover capacity
k_3 is the weight of the customer satisfaction on service quality
d_{oi} is the distance from the general warehouse to a front warehouse, and d_{ij} represents the distance from a front warehouse to a terminal customer
q_i is the freight volume of each front warehouse i
p_j is the demand quantity of customers j
f_i is the fixed operating cost of each front warehouse
$x_{\max}$ is the service level of the front warehouse
$\bar{t}_{ij}^1$ is the starting time point of the expected time window of customer j on warehouse i
$\bar{t}_{ij}^2$ is the ending time point of the expected time window of customer j on warehouse i
$\bar{t}_{ij}^3$ is the maximum acceptance time of customer j on warehouse i
L_{ij} is the distance between customer j and warehouse i
Q_k is the capacity of vehicle k, $k \in K$
Q_n is the capacity of vehicle n, $n \in N$
D_k is the driving distance capacity of vehicle k, $k \in K$
D_n is the driving distance capacity of vehicle n, $n \in N$

Decision variables:

x_{oik} binary variable, 1 indicates moving goods from the general warehouse to front warehouses i by vehicle k
x_{ijn} binary variable, 1 indicates moving goods from the front warehouse i to terminal customer j by electric vehicles n
z_{ij} binary variable, 1 if customer j is designated to warehouse i
y_i binary variable, 1 indicates the front warehouse being selected
s_{1i} is the customer satisfaction of customer i on distribution time
s_2 is the customer satisfaction of customer i on turnover capacity
s_{3i} is the customer satisfaction of customer i on service quality
st_{ij} is the customer satisfaction of customer i on distribution time regarding j
sx_i is the customer satisfaction of customer i on turnover capacity

Based on the description of the problem and the variables, an optimization model of logistics outlets under the two-stage new retail model is established.

$$\text{Min}Y = Z + \frac{\lambda}{S}, \tag{1}$$

$$Z = \sum_{i\in I}\sum_{k\in K} c_1 q_i d_{oi} x_{oik} + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_1 q_i d_{oi} x_{oik} + \sum_{i\in I}\sum_{k\in K} f_i y_i, \tag{2}$$

$$S = \sum_{i\in I}\left(k_1 \cdot s_{1i} + k_3 \cdot s_{3i}\right) + k_2 \cdot s_2, \tag{3}$$

$$s_{1i} = \sum_{j=1}^{m} \frac{st_{ij}}{|J|}, \quad \forall i \in I, \tag{4}$$

$$s_2 = \sum_{i=1}^{n} \frac{sx_i}{|I|}, \tag{5}$$

$$s_{3i} = \sum_{i=1}^{n}\sum_{j=1}^{m} \frac{y_{ij}}{|J|}, \quad \forall i \in I, \tag{6}$$

$$st_{ij} = \begin{cases} 1, & \bar{t}_{ij}^1 \le t_{ij} \le \bar{t}_{ij}^2, \\ 1 - \dfrac{t_{ij} - \bar{t}_{ij}^2}{\bar{t}_{ij}^3 - \bar{t}_{ij}^2}, & \bar{t}_{ij}^2 \le t_{ij} \le \bar{t}_{ij}^3, 0, t_{ij} > \bar{t}_{ij}^3, \end{cases} \tag{7}$$

$$sx_i = \begin{cases} 1, & x_i \le x_{\max}, \\ 1 - \left(\dfrac{x_i - x_{\max}}{\alpha x_{\max}}\right)^{\beta}, & x_i > x_{\max}, \end{cases} \tag{8}$$

$$\sum_{o\in r}\sum_{i\in I} x_{oik} = 1, \quad \forall k \in K, \tag{9}$$

$$\sum_{i=1}^{n} q_i x_{oik} \le Q_k, \quad \forall k \in K, \tag{10}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} p_j x_{ijn} \le Q_n, \quad \forall n \in N, \tag{11}$$

$$\sum_{i=1}^{n} d_{oi} x_{oik} \le D_k, \quad \forall k \in K, \tag{12}$$

$$\sum_{j=1}^{n} d_{ij} x_{ijn} \le D_n, \quad \forall n \in N, \tag{13}$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijn} = 1, \tag{14}$$

$$z_{ij} \le y_i, \quad \forall i \in I, j \in J, \tag{15}$$

$$q_i \ge \sum_{j\in J} p_j z_{ij}, \quad \forall i \in I, \tag{16}$$

$$\frac{L_{ij}}{v} \leq 0.5 + \left(1 - z_{ij}\right)M, \quad \forall i \in I, j \in J. \tag{17}$$

The objective function of this paper features cost minimization and customer satisfaction maximization, thus taking the minimum value of Z in formula (2) and the maximum value of S in formula (3) with no direct weighting and obtaining the minimum value by calculating the reciprocal of satisfaction degree. Because the order of magnitude between the two functions is not uniform, the transformation factor λ is introduced to transform the dual-objective model into a single-objective model, as shown in formula (1). To be specific, formula (2) shows the objective function of the total cost, which consists of the transportation cost from the general warehouse to the front warehouses, the distribution cost from the front warehouses to terminal customers, and the fixed operating cost of the front warehouses. Formula (3) represents the function of customer satisfaction, which is determined by three factors, namely, the distribution time (s_{1i}), goods turnover capacity (s_2), and service quality (s_{3i}), which can be obtained by formulas (4)–(6). In formula (4), st_{ij} represents the membership degree of terminal customers j to a front warehouse i in terms of distribution time, which can be calculated using formula (7). Similarly, the customer satisfaction of customer on turnover capacity involved in formula (5) can be calculated by formula (8), in which α and β in the function represent the customized coefficients for customers.

The above two objective functions have the nature of conflict. In other words, given a certain number of vehicles, if we reduce the transportation cost, distribution cost, and fixed operating cost, customer satisfaction will be improved. In this case, there is no conflict. However, the first objective function is a minimal one and the second objective function is a maximal one. Once one of them is optimized, the other objective would be impaired for certain.

Formula (9) presupposes that goods from a front warehouse can be delivered by only one vehicle. Formulas (10) and (11) guarantee that the goods loaded on the vehicle does not exceed its loading capacity. Similarly, formulas (12) and (13) presuppose that the length of the distribution line does not exceed the driving distance capacity. Formula (14) determines that each customer should be designated to one of the front warehouses. Formula (15) presumes that customers j can be served by the front warehouse i only when the front warehouse is selected. Formula (16) presupposes that the amount of goods in a front warehouse is not less than the amount demanded by its terminal customers. Formula (16) presets the load limit of the electric vehicle assigned to a front warehouse. Formula (17) presumes that the delivery time from a front warehouse to its customers shall not exceed 0.5 hours.

3. Algorithm Solutions

The immune algorithm is a new calculation method inspired by the antigens and antibodies of biological immune systems. It has the self-adaptive characteristic of obtaining the optimal solution of a multiobjective function by multiple mechanisms and thus can avoid being "premature" (converging to a local extreme value) to a large extent. In the immune algorithm, antigens correspond to problems to be solved and antibodies correspond to solutions. In the process of seeking a solution, the algorithm generates the optimal solution by simulating the biological immune system. The specific algorithm flow is as follows. First, input the antigen and expect the initial antibody to be generated and then calculate the affinity between antibodies; when the antibodies are varied to cause the metabolism of immune cells, new antibodies are generated by the selection, intersection, mutation operators, and immune operators. Under the influence mechanism of a set of judgment conditions, the antibody population of the immune algorithm is constantly adjusted, updated, and evolved, to search for the optimal antibody, that is, the optimal solution to the problem.

3.1. Antigen Input. The objective function and restrictive conditions of the problem are input as the antigen of the immune algorithm, and the solution to the problem is accordingly regarded as the antibody.

3.2. Initial Antibody Generation. In the algorithm, the immune system generates the initial antibody by searching memory cells, activating the memory function of the immune mechanism. After the memory mechanism obtains the solution (antibody) to a problem, it selects and retains a certain scale of high-quality individuals. In the process of antibody renewal, many excellent individuals are stored and the inferior ones are replaced in the memory bank.

3.3. Affinity Calculation. The affinity is a concept representing the relationship between an antigen and an antibody. The former is done with a fitness function (F) to extract the target function (Y). The latter suggests the similarity between antibodies, which is calculated with the following formula:

$$S_{v,s} = \frac{k_{v,s}}{L}. \tag{18}$$

In this formula, $k_{v,s}$ refers to the same digits of two antibodies and L refers to the length of the antibody.

Antibody concentration (i.e., the proportion of similar antibodies in the population) is calculated with the following formula:

$$C_v = \frac{1}{R} \sum_{j \in R} S_{v,s}. \tag{19}$$

In this formula, R is the total number of antibodies and $S_{v,s} = \begin{cases} 1, & S_{v,s} > T, \\ 0, & \text{otherwise} \end{cases}$, with T serving as a preset threshold.

3.4. Antibody Metabolism (Intersection and Variation). The metabolism of immune cells is caused by the intersection and variation of antibodies. In this paper, two randomly generated numbers are used as insertion points to

exchange customer information, and the same elements in the new antibody are successively deleted while the missing customer numbers are filled back in to form two new legal antibodies. To let the filial generation inherit more genes of the parental generation, an evolutionary reversal operator mutation method is adopted to generate a new gene string. After comparing the fitness of the new antibody and the original antibody, the one with higher fitness enters the next generation.

3.5. Promotion and Inhibition of Antibody Generation. In the process of finding the optimal solution, when the antibody concentration exceeds a certain value, the solution process will converge too early to get a globally optimal solution. The problem of too many antibodies or too many similar antibodies can be solved by controlling antibody concentration to prevent population degeneration.

Formula (19) calculates the antibody concentration. If individuals with higher concentration are defined as $A_1, A_2, \ldots,$ and A_t, the concentration probability of t individuals can be calculated by the following formula:

$$P_d = \frac{1}{N}\left(1 - \frac{t}{N}\right). \tag{20}$$

When $1 < t < N$, the concentration probability of the other $N - t$ individuals is

$$P_d = \frac{1}{N}\left(1 + \frac{t^2}{N^2 - N \cdot t}\right). \tag{21}$$

The individual selection probability (P) is made up of the fitness probability (P_f) and concentration probability (P_d):

$$P_f = \frac{1}{Y}, \tag{22}$$

$$P = \delta \cdot P_f + (1 - \delta) \cdot P_d. \tag{23}$$

In the above formulas, δ is the affinity coefficient, $\delta > 0$, and P_f and $P_d < 1$. Equation (23) suggests that the greater the individual fitness, the greater the selection probability, whereas the greater the individual concentration, the smaller the selection probability. Thus, not only can individuals with high fitness be retained, but also the diversity of individuals can be ensured, better avoiding premature convergence.

3.6. Group Renewal. Two antibodies can be randomly selected to undergo mutation according to the mutation probability set in the previous steps and then intersect with each other. Steps 3–5 are repeated until the termination condition of the algorithm is met.

3.7. The Judgment of Termination Conditions. When all parameters have met the termination conditions of the algorithm, the algorithm stops when the maximum number of iterations is reached and the algorithm ends when the best individual is the optimal solution; otherwise, return to Step 3 and repeat the above cycle.

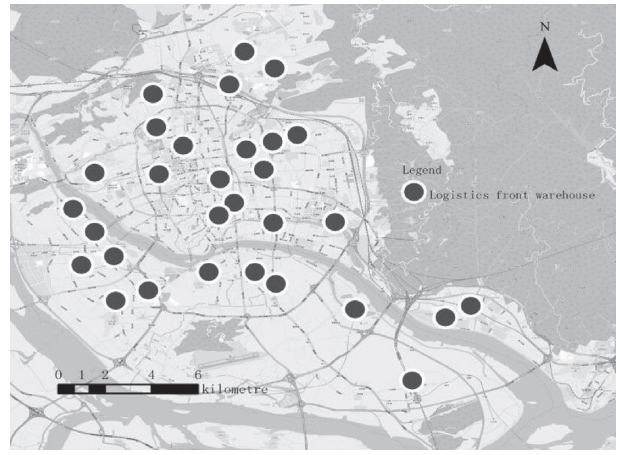


Figure 1: The layout of front warehouses.

4. Case Study

4.1. Data and Model Parameters. A Fuzhou-based Internet retail enterprise, https://www.pupumall.com, was made a case study here. Its general warehouse is located in Zeyang village, Nantong, Minhou County, Fuzhou, and it has 30 front warehouses available. Each front warehouse is responsible for the distribution of goods to several communities. The layout of the front warehouses is shown in Figure 1, and the quantity demand of each warehouse is shown in Table 1. Customers in 298 communities are covered by the service offered by https://www.pupumall.com. The parameters involved in the model are shown in Table 2.

4.2. Model Solution Output. The parameters of the immune algorithm are as follows: population size $N = 80$, crossover probability $P_c = 0.8$, mutation probability is 0.2, and iteration times are 100. The algorithm runs 80 times before finding the final solution, and the minimum value of the total cost is 960,630. Table 3 and Figure 2 compare the genetic algorithm and the immune algorithm, and the data suggest that each index of the immune algorithm is better than that of the genetic algorithm. The corresponding routes from the general warehouse to the front warehouses are shown in Figure 3, and the corresponding routes from front warehouses to customers are shown in Figure 4.

After 90 and 80 iterations, the genetic algorithm and immune algorithm reach the maximum fitness value of 0.762. Meanwhile, the average time needed for the immune algorithm to converge to get the final solution is 97.8 seconds, 9.2 seconds faster than the genetic algorithm. It can be seen that the immune algorithm has better convergence and stability. Besides, compared with the genetic algorithm, the immune algorithm used in this paper can facilitate getting the final routes of vehicles in a shorter time.

Table 1: Quantity demand of each front warehouse (kg).

q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}
2200	1800	1100	1700	3000	2000	2300	1400	500	1600
q_{11}	q_{12}	q_{13}	q_{14}	q_{15}	q_{16}	q_{17}	q_{18}	q_{19}	q_{20}
1900	400	1000	2400	600	3300	3800	2000	1500	2000
q_{21}	q_{22}	q_{23}	q_{24}	q_{25}	q_{26}	q_{27}	q_{28}	q_{29}	q_{30}
1300	1200	2100	2200	1900	1400	1200	1600	2700	1200

Table 2: Parameters involved in the model.

C_1	C_2	K_1	K_2	K_3	α	β	X
0.003	0.002	0.4	0.3	0.3	1	2	3000
t_{1ij}	t_{2ij}	t_{3ij}	λ	Q_k	D	L	N_k
0	0.5	2	10000	4000	100	1.5	1000

Table 3: Comparison between the genetic algorithm and immune algorithm.

Method	Total cost	Satisfaction value	Dual-objective	Iteration times	Max. fitness
Genetic algorithm	972,667	1.198	981,014	90	0.762
Immune algorithm	960,630	1.300	968,322	80	0.762

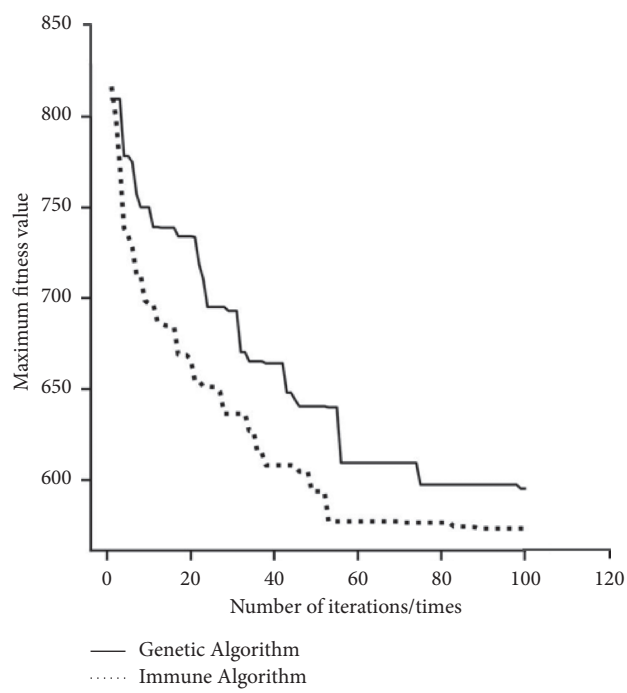


Figure 2: Fitness curve of immune and genetic algorithms.

We calculated the impact of two-vehicle models on the total cost of new retail logistics distribution. Table 4 shows that, with a 5-ton vehicle load, the total cost is 1.08% which is less than that of a 4-ton load, even with 20% fewer vehicles required, yet with better maximum fitness. It is therefore quite obvious that selecting vehicles with the same loading capacity as the demand capacity of the front warehouse will likely reduce the cost of logistics distribution.

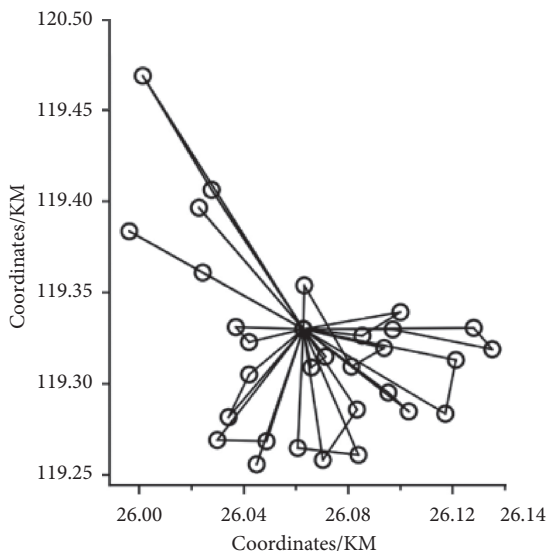


Figure 3: Returned routes from the general warehouse to the front warehouses.

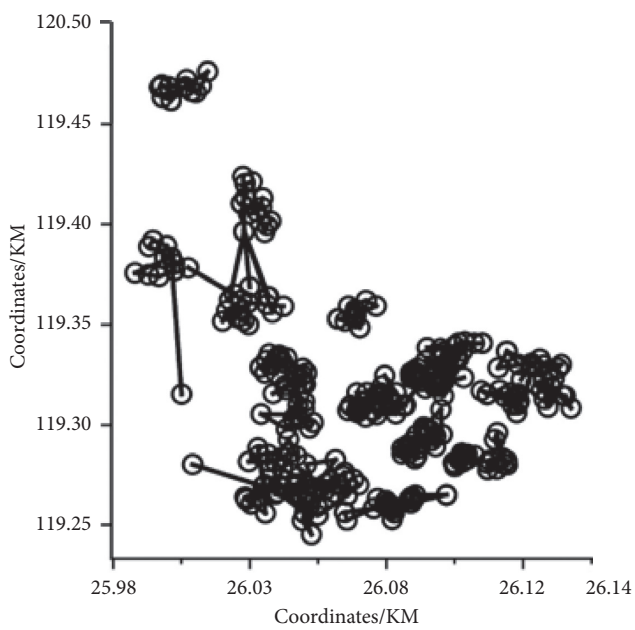


Figure 4: Returned routes from front warehouses to customers.

Table 4: Comparison of types of vehicles used in new retail network distribution.

Vehicle load	Total cost	Number of iterations	Number of vehicles	Maximum fitness value
4	960,630	80	15	0.762
5	950,219	92	12	0.763

5. Conclusions

This paper focuses on optimization problems of logistics nodes and vehicle paths that need to be solved jointly for new retail enterprises (such as https://www.pupumall.com) using the operation mode of online order plus front warehouse distribution and delivery. A dual-objective optimization model is therefore established with the minimum logistics cost and maximum customer satisfaction. Data of logistics distribution from the general warehouse and front warehouses of https://www.pupumall.com were made into a case study, and the immune and general genetic algorithms were used to obtain model solutions. The conclusions are as follows.

First, the immune algorithm adopted to solve the optimization model established in this paper is capable of effectively preventing the degradation of the genetic algorithm in the later stage and is superior to the genetic algorithm in search speed, fitness, and optimization effect.

Second, with the restrictive conditions in this paper taken into consideration, the case study shows that the cost of the corresponding logistics distribution scheme suggested by the immune algorithm is lower than the corresponding index value of the genetic algorithm, and the former is more practical. Besides, the logistics distribution vehicle model has a significant impact on the total logistics cost. The vehicle model with the same loading capacity as the demand capacity of the designated front warehouse is more likely to reduce the distribution cost.

In this paper, it is presumed that the number of end nodes is limited and the road traffic is smooth when conducting the optimization. It will be the goal of the follow-up study to obtain a more practical solution by expanding the search scope to fully cover the service area and account for the situation of traffic congestion and restriction.

Data Availability

The data used to support the findings of this study are included within this article.

Conflicts of Interest

The authors declare that there are no potential conflicts of interest.

Acknowledgments

This paper was funded by the logistics research team of Chuanzheng Communications College (205165).

References

[1] Q. W. Zhao, J. P. Zhao, and Y. Lin, "A city logistics network optimization model for large chain retailers under online–offline channel integration," *Chinese Journal of Management Science*, vol. 25, pp. 159–167, 2017.

[2] Y. Hu, "The optimization of logistics distribution network nodes in retail enterprises under 020 mode," *Business Economy Research*, vol. 8, pp. 72–74, 2018.

[3] J. Berger and M. Barkaoui, "A new hybrid genetic algorithm for the capacitated vehicle routing problem," *Journal of the Operational Research Society*, vol. 54, no. 12, pp. 1254–1262, 2017.

[4] A. T. Mulloorakam and N. Mathew Nidhiry, "Combined objective optimization for vehicle routing using genetic

algorithm," *Materials Today: Proceedings*, vol. 11, pp. 891–902, 2019.

[5] R. Cheng, "Research on distribution route optimization of B2C online supermarket based on data fusion algorithms under new retail mode," *Basic and Clinical Pharmacology and Toxicology*, vol. 125, no. 1, pp. 166-167, 2019.

[6] H. Pan, "Logistics distribution center location based on model optimization and immune optimization algorithm," *Electronic Design Engineering*, vol. 27, no. 10, pp. 78–81, 2019.

[7] C. Lu, J. Fang, and S. Fu, "A new equilibrium strategy of supply and demand for the supply chain of pig cycle," *Mathematical Problems in Engineering*, vol. 2020, Article ID 2093593, 13 pages, 2020.

[8] B. Wang, H. Zhang, J. Nie et al., "Multipopulation genetic algorithm based on GPU for solving TSP problem," *Mathematical Problems in Engineering*, vol. 2020, Article ID 1398595, 8 pages, 2020.

[9] T. Mareda, L. Gaudard, and F. Romerio, "A parametric genetic algorithm approach to assess complementary options of large scale windsolar coupling," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 2, pp. 260–272, 2017.

[10] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 517–530, 2017.

[11] J. Wang, M. Zhou, X. Guo, and L. Qi, "Multiperiod asset allocation considering dynamic loss aversion behavior of investors," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 73–81, 2019.

[12] X. Guo, M. Zhou, S. Liu, and L. Qi, "Lexicographic multi-objective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints," *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3307–3317, 2020.

[13] J. Qiao, F. Li, C. Yang, W. Li, and K. Gu, "A self-organizing RBF neural network based on distance concentration immune algorithm," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 276–291, 2020.

[14] L. Huang, M. Zhou, and K. Hao, "Non-dominated immune-endocrine short feedback algorithm for multi-robot maritime patrolling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 362–373, 2020.

[15] J. Chen, M. Zhou, and H. Zheng, "A novel radius adaptive based on center-optimized hybrid detector generation algorithm," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1627–1637, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9946458, 20 pages
https://doi.org/10.1155/2021/9946458

Research Article

The Business Process Reconstruction of Railway-River Combined Transportation Cloud Platform Taking China Container Export as an Example

Furong Jia, Lin Sun, Jiaxin Yuan, Yongping Li, and Qiang Huang

College of Information Engineering, Sichuan Agricultural University, Ya'an City, Sichuan 625014, China

Correspondence should be addressed to Qiang Huang; 19623556@qq.com

Received 4 April 2021; Revised 26 April 2021; Accepted 7 August 2021; Published 2 September 2021

Academic Editor: Chi-Hua Chen

In recent years, major large ports in China have realized the business informatization of rail-water intermodal transportation. However, the overall development level of intermodal transportation informatization has been restricted to a great extent due to the unbalanced development of intermodal transportation informatization in different regions, the rigid system architecture, the low degree of information sharing, and the lack of data management and analysis methods. Combined with the structure and business characteristics of intermodal transportation information systems, adopting cloud computing and Big Data technology, we propose an intermodal transportation information process with waybill as the information carrier and FPMS as the service fulcrum in this paper. Illustrated by the example of China's container export process, this paper explores a series of key technical issues in the cloud environment, such as application management, business information sharing, and Big Data processing, at different levels of the construction of the rail water transport cloud platform, combined with its business characteristics, and makes experimental analysis on the relevant models to verify the feasibility of the reconstruction of the rail water transport cloud platform. It can provide theoretical and practical support for the development of rail water intermodal informatization in China.

1. Introduction

At present, the information construction of railway-river transportation in China is all centered on ports, lacking overall planning, and there are many redundant constructions [1]. There are many differences in the information system construction architecture of various ports, which lead to serious information heterogeneity. At the same time, this information processing mode requires the railway to dock with different ports one by one, which will consume a lot of unnecessary time when the railway information is docked with the port business process, and the port also lacks effective expectation and control of the business growth [2]. With the continuous emergence of new combined transport services and the rapid expansion of the scale of services, the entropy value of the combined transport information system scattered in various ports will increase rapidly and cannot bear the burden of business development [3]. Che Kingsley chenikwi establishes the possibility of sustainable multimodal transportation by integrating different transportation methods to solve the problem of transportation fragility and promote the development of effective and efficient transportation networks in sub-Saharan Africa [4]. The loading and unloading operations of the Port of Rotterdam in the Netherlands are all managed and controlled by computers, and the development and construction of logistics parks are actively carried out to give full play to the logistics advantages of the port. The British railway transportation system actively cooperates with the port transportation system to comprehensively improve transportation quality and efficiency.

There are relatively rich theoretical studies on domestic and foreign ports and railway river transport and related issues, mainly for container ports, focusing on the improvement of railway river transport quality and efficiency,

the optimization of railway river transport, and the integration and optimization of railway-rail transport models and algorithms. The railway-rail transportation standardization service system has been established, but there is little research experience on the railway river transportation information platform. This article focuses on verifying the feasibility of rebuilding the railway river transportation cloud platform, and provides theoretical and practical support for the development of the railway river.

The main work of this paper is summarized as follows:

(1) This paper proposes a cloud platform for railway inland transportation, which applies information technology to the "centralized" mode of informatization. Adopting the information management mode of "large concentration" can make full use of virtualization technology, give play to the advantage of resource intensification, and centralize the management of intermodal transportation information system and business data distributed in various ports. In the large centralized management mode, the powerful resource management ability of cloud platform can be effectively used, so that small- and medium-sized ports can get rid of the limitation of intermodal information, scientifically and reasonably manage and control the cost, properly deal with the management complexity and other problems, and have the same level of intermodal information as large ports in a short time. Compared with the decentralized information layout, the main advantages of the "large concentration" mode are shown in Table 1 [5].

(2) The integration and reconstruction of the overall business and the realization of hierarchical management of the process can effectively simplify the information interaction process, improve the efficiency of information integration, and more effectively manage the combined transportation information system and the business data distributed in each port [6].

(3) The business operation simulation process of Petri net model is established, which improves the business efficiency of MTR docking and reduces the cost. Container transportation process is characterized by dynamic, uncertainty, and complexity. Compared with the existing process modeling methods, such as event-driven process chain, data flow diagram, IDEF and Petri net, the Petri net can analyze the performance of the process concisely and efficiently. The comparison of methods is shown in Table 2 [7].

2. The Information Layout of the Cloud Platform of Railway-River Combined Transport

2.1. Intermodal Transportation Information Distribution Status. At present, China's railway-river transport informatization is constructed in a decentralized way centering on ports [8]. Each port builds its transport information system according to its business development, and integrates information with other transport participating institutions, as shown in Figure 1.

As shown in Figure 1, large ports have invested a lot of energy in planning their own business and building a relatively complete intermodal information system [9]. Small- and medium-sized ports are incomplete due to manpower and financial constraints. The railway-river transport information platform is a complex giant system. In addition to the characteristics of the general information system, it must be integrated, dynamic, and real-time. We found that the current disparate intermodal information layout and the isomerization of information systems are serious through the investigation of the intermodal information system of Lianyungang, Ningbo Port, and other ports. Problems such as resulting in a low level of informationization in China's intermodel transportation, low level of information sharing and interaction, and disproportion service capabilities [10].

2.2. Advantages of "Centralized" Information Management Mode. According to information layout problems of intermodal transportation at present, this paper argues that the railway-river intermodal cloud platform should be involved in the business of the construction of railways, ports, and other departments jointly. Using the centralized information management mode to construct the railway-river intermodal cloud platform can make better use of virtualization technology, give full play to the advantages of resource-intensive, and realize the centralization of distributed port transportation management information system and business data [11]. The main advantages of the "centralized" mode are as follows [12]:

(1) All port business processes are standardized and unified.

(2) It makes data mining easy because of unified data storage, real-time sharing of information between ports, and access is restricted by the permission system.

(3) The intermodal management agencies make one investment in the port, centralized operation, and maintenance, so the maintenance cost is low.

(4) Information is exchanged through cloud service mode, and service resources can be shared among tenants with a high utilization rate. Renting services on demand instead of buying software with low cost; unified service port, access can be anytime, anywhere, large service capacity, and high reliability.

(5) Flat operation and maintenance mode, highly shared operation and maintenance personnel, high utilization, and efficiency.

(6) Use unified service portal, join port more, and promotion efficiency is higher.

(7) Centralized risk control ensures service and network communication quality through high reliability of the cloud and adopts a unified application security strategy to ensure information security.

Table 1: Comparison of technical advantages between decentralized layout and centralized mode.

Indicators/models	Distributed layout	Large concentration mode
Business standardization	Port business process is not unified and data management is difficult	All port business processes are standardized and unified
Information sharing	It is difficult to realize information sharing among ports, and the data are scattered in each port, so it is difficult to extract and mine the data	The data can be stored in a unified way, the information between ports can be shared in real time, the reading is limited by the permission system, and the data mining is easy
Construction cost	Repeated investment and high maintenance cost of port	One-time investment and centralized operation and maintenance by combined transport management organizations
Application development and update difficulty	The "chimney" structure is adopted, and the system renewal cycle is long	Using the virtual software package, the system update cycle is short
Information exchange mode	Information interaction based on system has low resource utilization, high interaction cost, low efficiency, and poor reliability	Information is exchanged through cloud service mode, and service resources can be shared among tenants with high utilization rate; it is cheap to rent services on demand instead of purchasing software; the service port is unified and can be accessed anytime and anywhere, with large service capacity and high reliability
Operation and maintenance mode	Vertical operation and maintenance mode, serious redundancy of operation and maintenance personnel, low utilization and efficiency	Flat operation and maintenance mode, highly shared operation and maintenance personnel, high utilization and efficiency

Table 2: Comparison of process modeling methods.

Attributes	Petri net	EPC	DFD	IDFE
Dynamism	Strength	Strength	Weakness	Weakness
Description method	Graphical	Structured and graphical	Graphical	Structured
Process simulation	Supported	Unsupported	Supported	Unsupported
Modeling direction	Process oriented	Blend	Process oriented	Process oriented
Comprehensibility	Strength	Strength	Strength	Weakness
Expressive ability	Good	General	Good	General

2.3. Information Layout Mode of Intermodal Cloud Platform. Combining with the current informatization layout disadvantages of intermodal transportation, on the basis of making full use of the advantages of cloud platform intensive resource management, the centralized construction of software and hardware resources and the unified migration of applications are realized, forming a highly centralized information layout mode, as shown in Figure 2.

As shown from the information layout, management mode is based on the cloud platform which breaks the platform independence of each other between the pattern of using virtualization technology to restructure and computing resources to each according to its need. This mode will be dispersed in the ports intermodal business application migration to the cloud to intensive management and by the unified entrance of foreign service, to provide one-stop integrated transport business information service, at the same time also can use the cloud application service mode for users effectively and personalized business needs rapid customization resources or business environment. The cloud platform based on information centralization management mode can make full use of the cloud scale effectively due to numerous ports in China. And it can greatly reduce the information cost of the whole industry, especially for many small and medium-sized ports, which can through a variety of terminal direct access and use the cloud computing resources platform, achieving zero cost information construction investment [13].

3. Business Process Restruction

It is necessary to further extract the business information based on the transportation organization process, combine the cloud platform system oriented to the centralized transport information management, build the business support system, build the Big Data system combined with the Big Data processing mode, and take the centralized box import and export as an example to reconstruct the operation information process of railway-river combined transport.

3.1. Business Support System Architecture Design. The business support system mainly aims at six major user groups: cargo owners/through transport operators, railway, port, shipping companies, one customs clearance, two inspections and other government agencies, as well as container truck distribution companies/insurance banks to provide application information services [14]. The information demand relationship between these six major user groups and the business support system needs to optimize the following

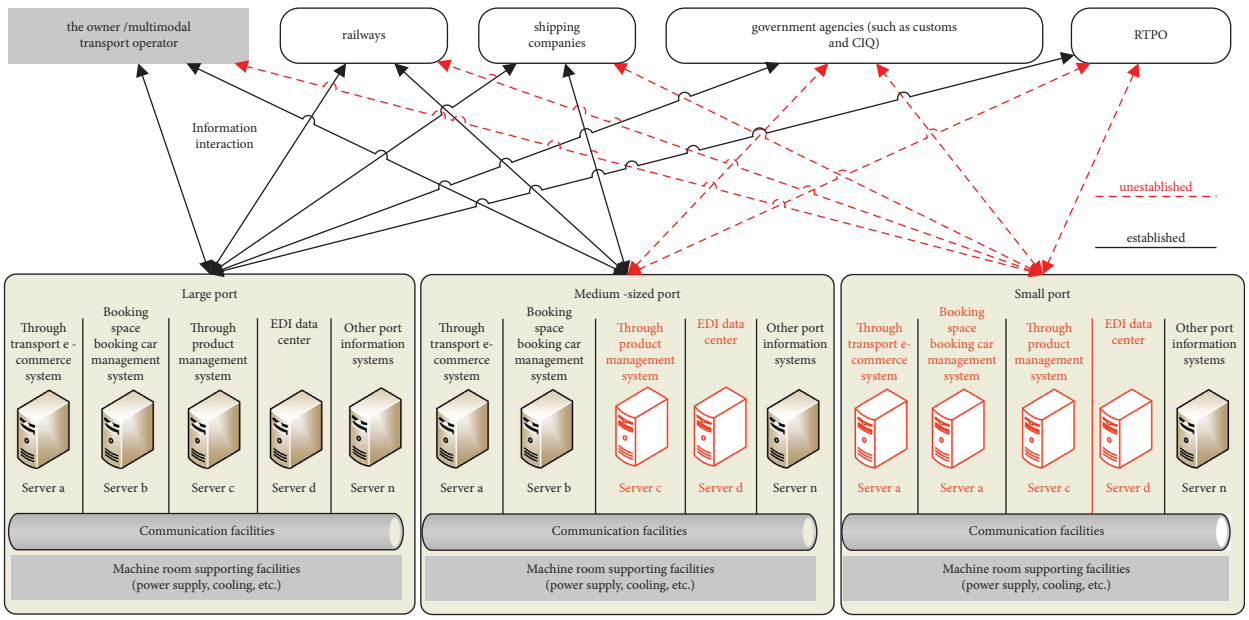


Figure 1: The port-centered information model of railway-river combined transport.

business on the basis of the existing through transport information system in terms of functional demand: the first is to build one-stop through transport services with customers as the center; the second is to enhance the sharing level of information on through transport between railways and waterways; and the third is to realize the management and analysis of Big Data on through transport.

The main function of the business support system is to integrate the current decentralized intermodal transport information system into an organic whole on the hot-water intermodal transport cloud platform. After configuring service instances and access rights according to customer needs, the system will be delivered to users in the SaaS service mode as shown in Figure 3.

According to Figure 3, the business support system mainly includes five parts: security framework, unified service portal, intermodal Big Data application, information sharing pool, and third-party application pool. The functions are as follows:

(1) Security framework: UGAC is an SaaS service mode based on cloud environment, which is responsible for unified access control of applications to eliminate the application security bucket effect in the business process.

(2) Unified service portal: The core of business support system is unified service portal, which is composed of intermodal e-commerce system and freight process management system to realize intermodal business.

(3) Intermodal Big Data application: The application of intermodal Big Data mainly includes the system instance pool related to intermodal business, in which the intermodal e-commerce platform provides customers with information query, consignment handling, cargo tracking, and other related information services in various transportation operations. Intermodal business system is mainly responsible for the planning, implementation, and management of intermodal logistics process. Intermodal marketing management system is mainly responsible for the standard formulation of intermodal products and freight rates. The collaborative office system is mainly used to ensure the normal daily management of the intermodal transport management department.

(4) Information sharing pool: The information sharing pool is mainly responsible for heterogeneous application integration and information sharing. It is oriented to the needs of intermodal transport participants and realizes business collaboration and information interaction between different transport modes. It is used for data exchange with private cloud applications, such as railway, large ports, government agencies, and other third-party applications.

(5) The third-party application pool: The third-party application pool is mainly composed of the business systems of the joint transport participating institutions, mainly including the business systems related to the joint transport of the port and railway departments, the small ports with low degree of information, and the fast provision of complete management information services related to the port station, wharf, and yard by means of leasing, so as to achieve zero information investment.

3.2. Technical Framework of Data Support System. Data support system is mainly used to realize the management and secondary utilization of Big Data in combined transport. To ensure the flexibility and expansibility of data management, data management and application are independent. The data support system first precipitates and sorts out and

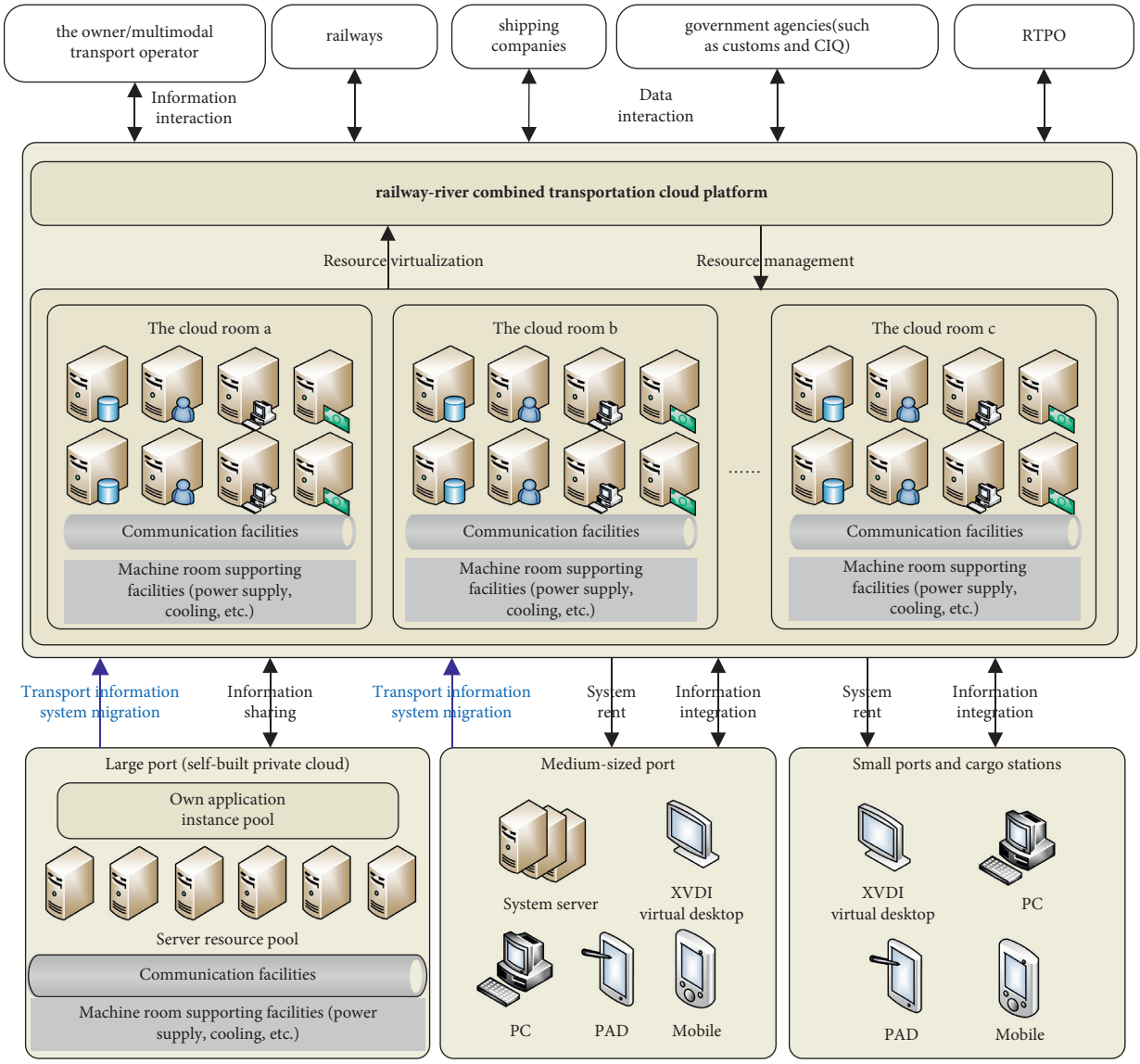


Figure 2: The centralized informatization mode of railway water cloud platform.

stores all through transport business data so as to form a data resource pool of the fragmented business data, then mines the precipitated data through the application of Big Data of through transport and delivers the results to the application of the business support system in the mode of data and service (DaaS) for consumption. These Big Data applications can, on one hand, react to the intermodal transportation applications that generate data and optimize the business itself; on the other hand, they can enhance the user experience and quality of service and provide valuable value-added data services to the intermodal transportation participants and managers in other industries, thus enhancing the core value of the entire platform.

According to the design of cloud platform, the data support system based on DOA can not only retrieve and process Big Data efficiently but also decouple data management and application requirements, and provide flexible data services for users and managers of cloud platform. The technical framework design is shown in Figure 4.

As shown in Figure 4, the data support system is composed of client and cloud service side. The client side realizes the data interaction between the user and the data support system through protocol adapter, and the system interaction between the customer and the data support system is realized through the business application pool. The cloud service side mainly includes three components: the DOA data management center, the business application pool, and the Big Data analysis center, respectively, corresponding to the registration management, generation, and analysis process of the Big Data for intermodal transport. The functions are as follows:

(1) DOA Data Management Center: It is composed of a data storage terminal and a data registration center, in

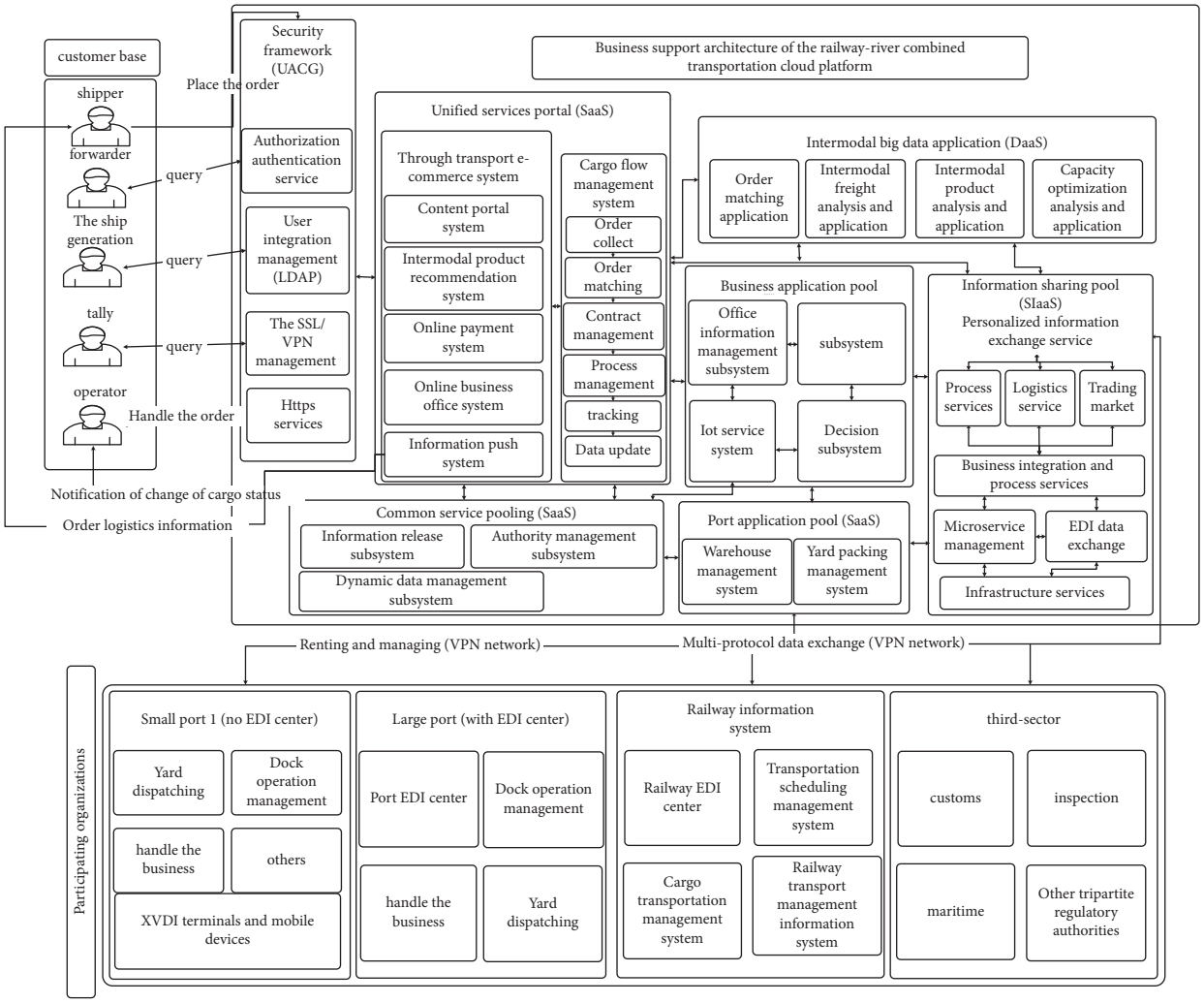


Figure 3: Business support architecture.

which the data storage terminal adopts an industrial cloud-distributed storage system to realize the unified storage and management of application data; the data registration center carries out unified management and operation of the data stored by the client and the service terminal through data registration, dynamic data aggregation of the client, and other technologies. The data are generated as needed in the metadata documents in the DOA Central Administration and stored in the metadata database in the registry. When a Big Data application needs to load a large amount of heterogeneous data for analysis, the DOA Central Administration retrieves the metadata information needed to load the data upon request and returns it to the Big Data application. A Big Data application system shall analyze the location and access interface of the data source from the metadata and obtain relevant data services. The data analysis results of Big Data applications are also stored in the data management center to form the corresponding data services (DaaS) for use by users and multimodal transport business applications. DOA data management center uses Spark protocol to realize the offline and online processing of combined transport data, and uses Hive data warehouse to manage the data information in distributed system. The better performance of Spark solves the problem of complicated data information in the course of operation and development of combined transport industry, and improves the efficiency of data processing. Hive-SQL realizes the transformation of MapReduce program, and improves the efficiency of data extraction and data cleaning. The performance comparisons of Spark with other data analysis platforms are shown in Table 3, and the characteristics of Hive are shown in Table 4.

(2) Business application pool: The business application pool can not only ensure the normal development of railway-water combined transport business but also be the producer of diversified combined transport Big Data from the perspective of data. These data are centrally stored and managed in the data support system, isolated by cloud service mode, and provided to combined transport Big Data applications as required.

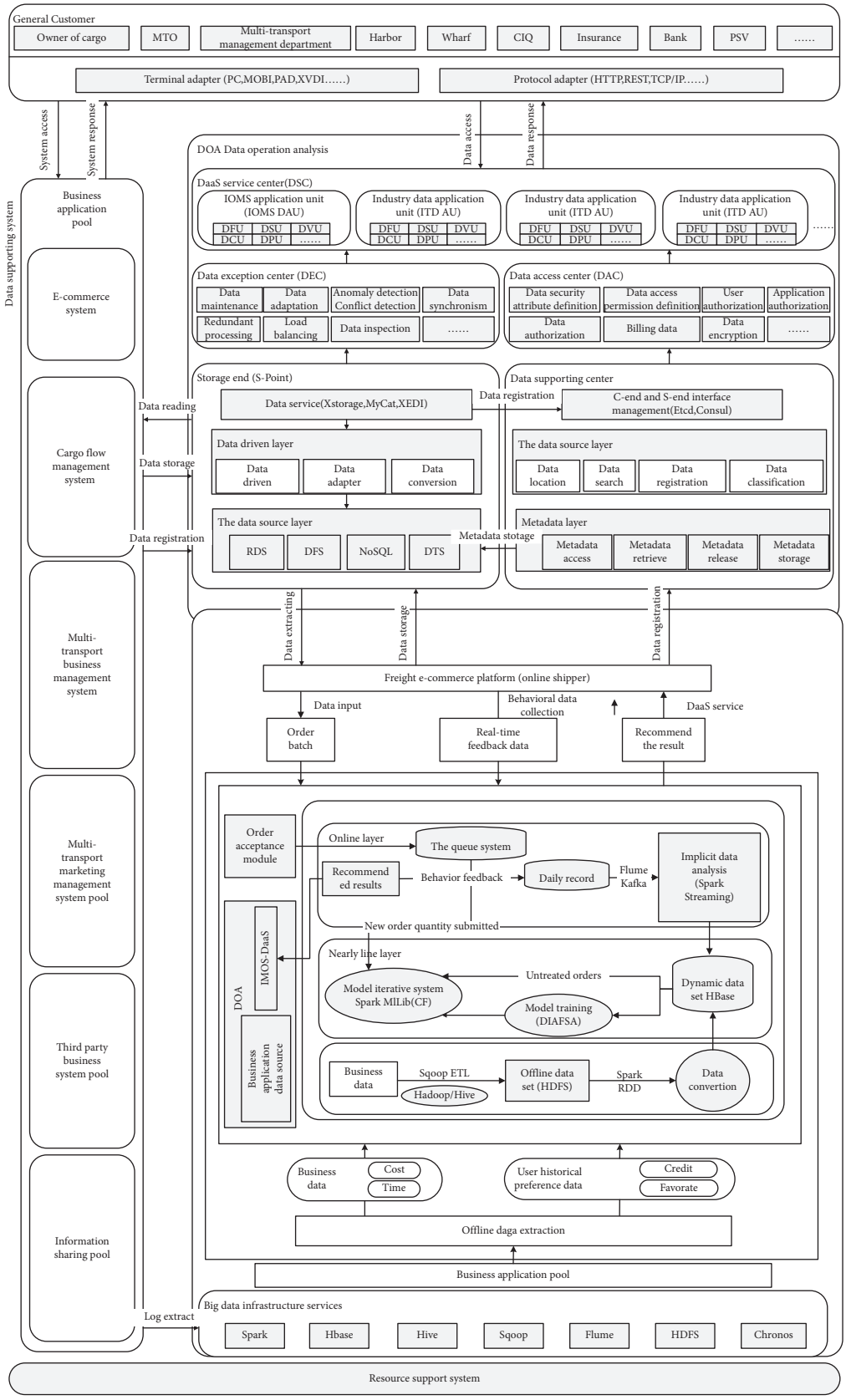


Figure 4: Data support architecture.

(3) Big Data analysis center: The analysis center consists of a Big Data application pool, which is a collection of data processing units (ITDPUs) that are independent of each other and are designed and deployed according to application requirements, and a Big Data infrastructure service pool. ITDPU applies to DOA and extracts the data, then loads the data into ITDPU for data analysis, and finally returns the results to DOA administration center for business applications and industry users in the DaaS mode. The Big Data Infrastructure service includes seven parts: Spark, Hbase, Hive, Sqoop, Flume, HDFS, and Chronos, to serve the intermodal Big Data application pool. Spark realizes the offline and online processing of intermodal transport data, Hive realizes the cleaning of unstructured fragment data, HDFS realizes the storage of Big Data and becomes the offline data source of dynamic data set, Sqoop realizes the collection and processing of Big Data, Flume realizes the collection of user feedback data and log data, Hbase realizes the storage of dynamic data set, and Chronos regularly schedules the offline data transmission module and realizes the scheduling management of data processing system.

The three components of the data support system cooperate with each other to form a closed-loop ecosystem of Big Data from generation, storage, management, to extraction, analysis, feedback, and service. Therefore, in the data support system, DOA data management center, business application pool, and Big Data analysis center are indispensable.

3.3. Information Sharing Mechanism of Intermodal Transportation Cloud Platform. At present, the information sharing technologies adopted by ports, railways, and other core transport agencies mainly include the following:

(1) Electronic data interchange (EDI): EDI standardizes and formats the exchanged information according to the agreed protocol (EDIFACT, SOAP, etc.), and carries out data exchange among the computer network systems of trading partners through mail servers, FTP, Message Queue (MQ), and other data transmission systems, which can effectively solve the inefficient problem of paper information transmission [15, 16].

(2) Service oriented architecture (SOA): Its essence is enterprise application integration (EAI) technology that realizes information exchange between heterogeneous systems [17]. The SOA component model realizes the business information interaction between heterogeneous systems by defining standardized interfaces between different services, which is characterized by loose coupling, coarse granularity and transparency [18, 19].

(3) Enterprise service bus (ESB): The ESB takes services as the basic constituent unit and realizes service coordination among services through messages, and completes relevant business coordination [20]. The ESB can not only reduce development and maintenance effort, save costs, and improve system scalability but also better handle the heterogeneity between different technologies and protocols [21].

4. Information Sharing Model Based on MSOA

Microservice architecture (MSA) is a typical application architecture, which can be directly managed by the container cloud environment of resource support system, with strong flexibility and robustness. Using MSA to build the information sharing model and using microservice to replace the heavyweight bus structure of traditional SOA, information sharing components can be decentralized and more completely decoupled to improve the performance of message processing.

This paper proposes to take microservices as the carrier to virtualize and encapsulate the logic of the information sharing process, such as routing and message parsing. And combine them dynamically according to the information sharing needs to form a microservice oriented architecture (MSOA) information sharing model, which is the core of the information sharing pool of the business supporting system. This model can make full use of the advantages of the integrated application management system and make the information sharing function available to users in the cloud service mode of information Integration as a service (SIaaS), to realize the on-demand sharing of intermodal transportation information.

Different from the current bus sharing technology of information of combined iron-water transportation and intermodal transportation, the information sharing model based on MSOA uses fine-grained microservice unit to decouple information interaction, and uses container virtual image for standardization to form a lighter and more flexible decentralized structure than the traditional cloud platform based on the virtual machine. In addition, the MSOA Shared unit can be distributed and managed under the same management system. It avoids the high cost caused by the deployment of a special server required by the current information sharing technology and makes full use of its scaling mechanism to improve the concurrency performance of the information sharing system. All because it is fully compatible with the intermodal application structure.

4.1. Model Framework Design. The MSOA sharing model is built on the application pool managed by the resource support system, which encapsulates and combines the information sharing business with the microservice as the unit, calculates resource allocation on demand under the resource support system, and provides it to the freight flow management department. Finally, the SIaaS mode is adopted to uniformly deliver to service consumers for use. The intermodal transportation participating organizations and customers only need to rent relevant shared services according to business needs to realize information sharing. The architectural design is shown in Figure 5.

Table 3: Performance comparison between spark and other data analysis platforms.

Data analysis platform	Hadoop	Spark	Storm
Data processing model	Batch processing	Batch and real-time processing	Real-time processing
Delayed	Higher	Second level	Real time
Cluster support	Thousands of nodes	Over 1000 nodes	Good
Throughput	Good	Good	Quite good
Applicable scene	Mass computation with low timeliness	Large data blocks need high-effective small-batch computation	Real-time analysis of small data blocks

Table 4: The characteristics of hive.

Characteristics	Hive
Scalability	Computational/scalability capabilities are designed for very large data sets
Metadata management	Provide unified metadata management
Malleability	Hive supports user-defined functions that users can implement according to their own needs
Fault tolerant	SQL can still execute if the node has a problem

According to Figure 5, the MSOA architecture consists of the management core of the Intermodal Transportation Integration Unit (ITIU), which is divided into the management layer and the business layer, in which the management layer is mainly responsible for the interface registration and process management of the ITIU and consists of the following three systems:

(1) Microservice governance system. Microservice governance system is responsible by MSA and application registration center of resource support system, which provides registration management of shared interface of intermodal application and service composition of ITIU. MSA registry is based on application, without the concept of SOA bus, and only provides service query, load balancing and fault-tolerant mechanism. The direct communication between services does not need to rely on any service bus, which can effectively decentralize the information sharing model.

(2) EDI message switching system: Unlike the ESB, the protocol supported by the MSA is not perfect, and the RPC-based synchronous invocation approach cannot guarantee performance stability when invoked remotely across regions and is only suitable for information integration between homogeneous CNA applications; and for heterogeneous information integration across departments (for example, with ports and railways, and government departments), EDI is currently the most widely used data standard in the combined transport industry. In the MSOA model, EDI is no longer used as an information integration bus, but only as a message exchange module, which is only used to realize the interdepartmental and interregional information integration.

(3) Process management system: After the integration of MSOA application information, it needs to be combined according to the shipping business process used by freight process management system (FPMS). Process management mainly relies on workflow system to provide cross-application business process integration support, and choreographs the underlying application components and services after being integrated by MSA.

Therefore, for the MSOA transport information sharing model architecture, the management's microservice governance system, EDI message exchange system, process management system, and business layer are of great importance.

4.2. ITIU Design and Implementation. In the freight business process, the shipping business is the beginning, determine the order data, complete the forwarding. At this time, railway and waterway departments also need to carry out specific operations to implement specific transport events and procedures. The business process of consign apply integration unit (CAIU) is defined as follows:

(1) FPMS applies to the MSOA Shared model for shipping.

(2) The MSOA shared model invokes the CAIU service. It is independently based on the current capacity status, judges the result of the application, and sends it to the railway department, which will approve it.

(3) Finally, CAIU transmits the relevant results to FPMS (for customer inquiry) and the relevant waterway departments (to arrange relevant plans).

The cross-system interaction process of CAIU is shown in Figure 6.

As can be seen from Figure 6, CAIU consists of four roles and four interaction processes. From the perspective of workflow, the main process includes one judgment node and one concurrent node. CAIU's main four interactions are as follows:

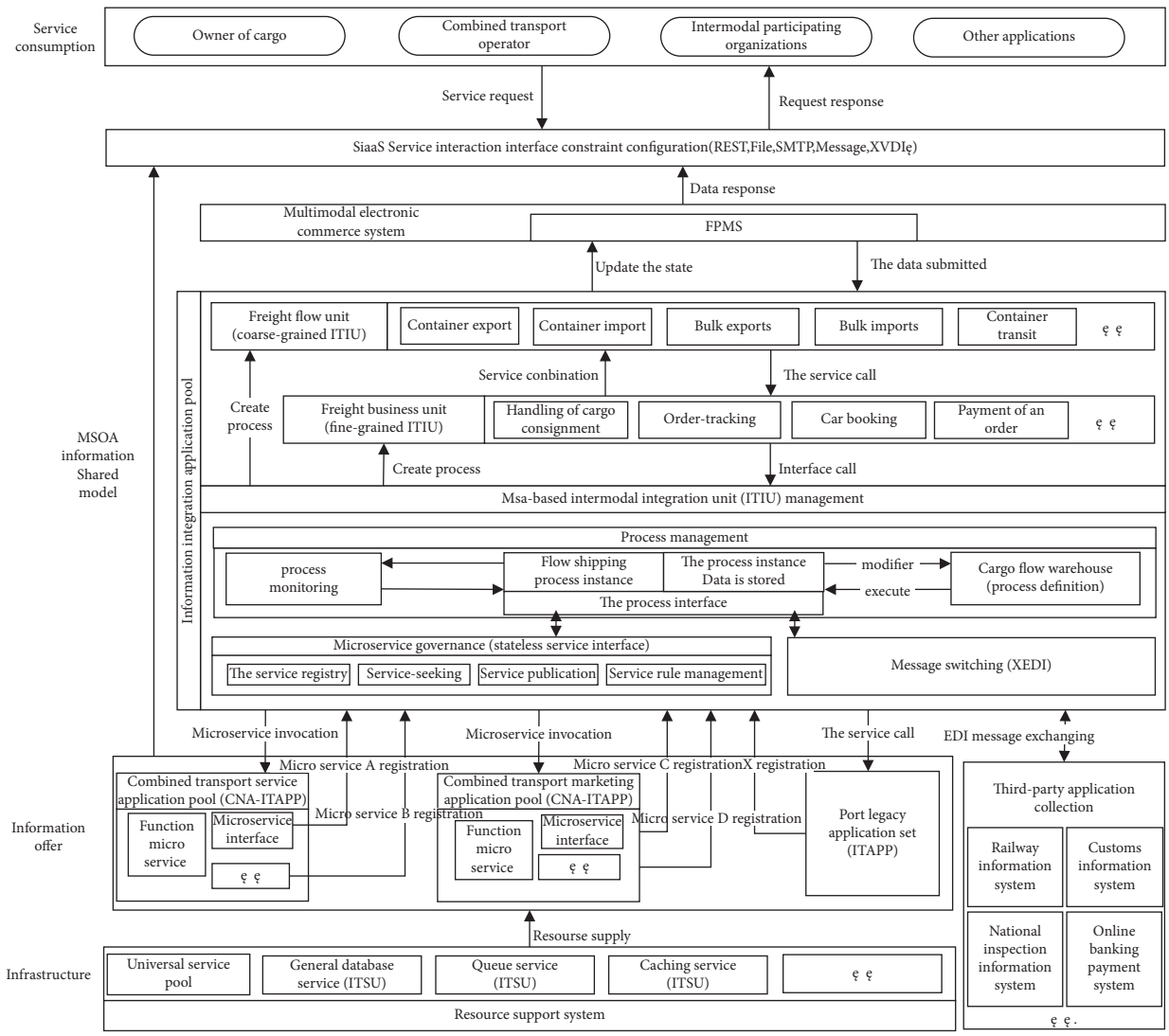


Figure 5: MSOA intermodal information sharing model framework.

(1) Order data submission interaction: FPMS submits to CAIU the basic cargo information, shipper information, consignee information, and shipping order information of the order.

(2) Vehicle application verification interaction: CAIU uses the intermodal freight management system to effectively compare the information of order goods and other data against the status of freight train capacity in the database, to judge whether the requirements for vehicle application are met or not.

(3) Order approval management interaction: based on the EDI system, CAIU transmits the verification result of vehicle application and the basic information message of order to the railway freight management department. With the assistance of the railway freight management information system, the railway management personnel carries out inspection, and there are no errors. Then CAIU replies the approval message of the application to CAIU.

(4) Order result interaction: CAIU sends the result of the order approval to FPMS and communicates with the shipper/intermodal operator for subsequent operations. If the consignment application is approved, send the order information to the freight station, make various plans in advance, communicate with the water transport department, and timely submit the relevant documents and approval results of the consignment.

4.3. Freight Flow Restruction Ideas. At present, the business process of combined transport is still faced with loads of problems, such as poor service quality, low concurrency, and lack of unified logistics information carrier. We established a unified business process model based on the transport cloud platform which can provide the whole information needed by users to improve business process efficiency [22].

The business process of combined transport is mainly comprised of an intermodal e-commerce system and the

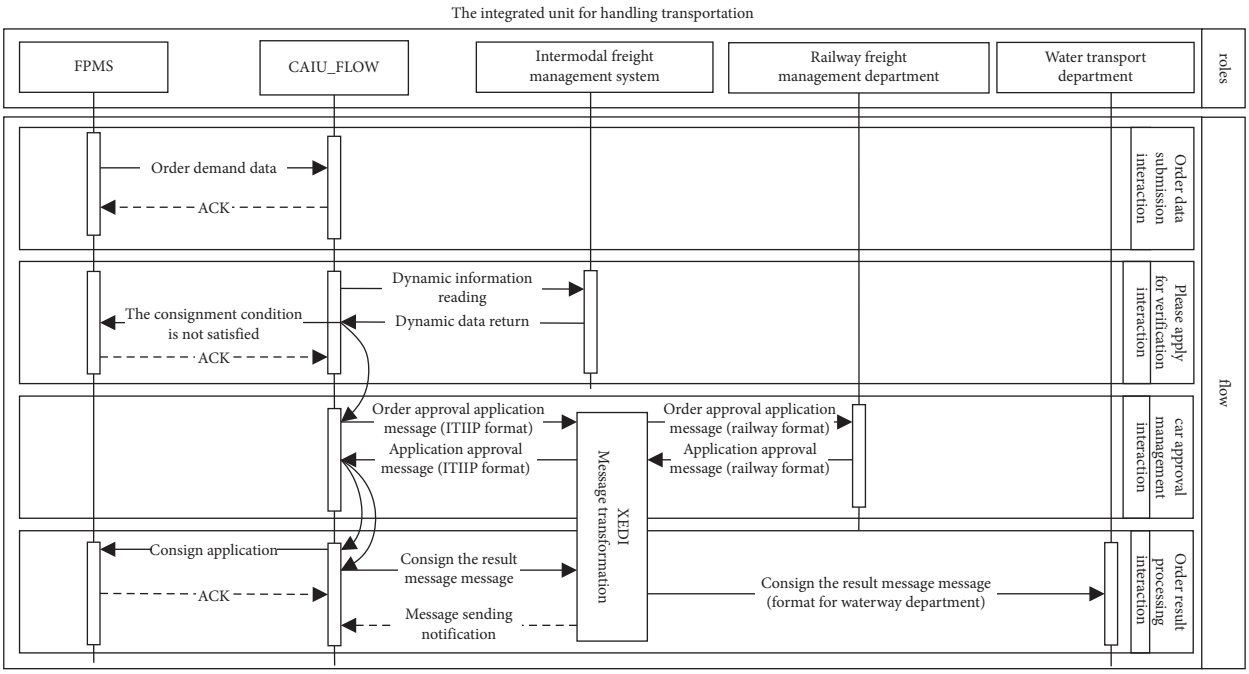


Figure 6: CAIU integrated unit design.

Freight Process Management System (FPMS). The former is an online C2C freight business service platform provided by the intermodal transportation operator, who provides information service with the help of the participating institutions as a merchant. The shipper, as a buyer, applies for the business through a unified port online, including contracts management, orders tracking, and payment. FPMS adopts the order driven model (ODM) to manage the processes of combined transport logistics [23].

We propose a new intermodal transportation information process based on FPMS. They are handed over to FPMS for process management and information integration all after the consignment orders are collected by the e-commerce platform. From a macro perspective, the whole business process is divided into the following four stages:

(1) Order demand submission stage: after authentication and authorization, the shipper submits his/her consignment demand to the intermodal e-commerce platform. After collecting and collating the order, it transfers them to FPMS.

(2) Orders allocation: after obtaining the batch order from the e-commerce platform, FPMS automatically allocates them using the order-matching Big Data application. The system calculates the most optimal target operator and recommends it to the shipper. When the shipper confirms the targeted operator, it goes to the next stage.

(3) The business management of logistics: when the shipper finishes signing contracts, the intermodal transportation operator shall make the plan in accordance with the contract requirements. They will book a car and a ship, contact the customs or third parties online, and will feedback the item's status information through the interface to the e-commerce platform until the delivery is complete.

(4) Logistics management stage: after finishing online payment confirmation, service feedback, dispute arbitration, and service evaluation, and so on, it provides data support for subsequent decision analysis according to the completion status of the logistics business.

4.4. Information Flow Design of Centralized Box Outlet Based on FPMS. There is a bad difference between import and export business processes. Take the centralized container export process as an example. The design of the entire export process includes the freight business and freight transport process. It adopts FPMS intermediary guarantee transaction mode, which is divided into the following four steps mainly: (1) order submission; (2) order matching; (3) contract signing; and (4) payment confirmation.

The cargo transport process is roughly the same as the current process, which is mainly divided into the following nine links [24]: (1) shipment handling; (2) vehicle booking; (3) tally and loading; (4) railway transport; (5) transshipment of goods at the port; (6) customs inspection; (7) export preparation; (8) loading; and (9) delivery.

The traditional railway-river combined transport information process is complex and not unified. FPMS combines the business flow and logistics, and uses an electronic waybill, to realize the paperless cargo transport and online one-stop service. The combined transport information process based on FPMS mainly has the following advantages:

(1) Business flexibility: FPMS can automatically recommend co-carriers for shippers to choose according to the order information of shippers.

(2) Business security: Shippers, intermodal operators, railways, ports, and so on, sign multiparty insurance, unified claims, and effectively reduce the risk of transport.

(3) Convenient transportation: The traditional railway and sea freight agents are integrated into the combined transport agent, and the shipper only needs to communicate with the combined transport agent, while the combined transport agent conducts business negotiation with the port and railway through FPMS;

(4) Real-time logistics information: Railway and marine information will be uploaded to FPMS in real time, and the corresponding information will be automatically made available to each participant.

4.5. Modeling of Container Export Information Flow Based on Petri Net. Petri net is a structured mathematical language, which can not only describe the process graphically but also convert the graphics into mathematical calculations to evaluate the behavior of the system. The object-oriented traffic system simulation model should be constructed, and the object-oriented idea is combined with Petri net to realize the high reusability and operability of the model, and reduce the scale of the model, which is suitable for all activities, including logistics system. A formal presentation Petri Net is following [25]:

$$PN = (P, T, F, W, M_0),$$
$$F \subseteq (P \times T) \cup (T \times P), \tag{1}$$

$P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places (Place), which represents the set of nodes and departments where transportation tasks start, end, or information flow, n is the number of finite places.

$T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions, that is, the set of the execution of transportation tasks or the process of information transmission, and m is the number of finite transitions.

P and T satisfy condition $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$.

W: $F \longrightarrow \{1, 2, 3, \ldots\}$ is the weight function of the directed arc.

M_0: $P \longrightarrow \{0, 1, 2, 3, \ldots\}$ is the initial mark.

Through the Petri Net model, we use OOPN to simulate the flow of container export information. According to the model composition of OOPN, the system composition object and structure are determined by business analysis, and the information flow of container export is modeled according to the following steps:

(1) Divide the composition of information flow system of FPMS into four categories of objects.

(2) Determine the behavior of the class object itself and its relationship with other objects.

(3) Construct the internal behaviors of four types of objects with simple Petri nets.

(4) Based on (3), an object message queue is built and an input/output gate is used to represent the external interface of the object.

(5) Connect corresponding transitions in internal Petri nets of various objects with message teams.

At the same time, to simplify the modeling process, the following assumptions were made for the FPMS process system before model construction:

(1) Railway freight station can provide uniform service within a period of time.

(2) All transport-related departments and personnel related to freight can keep in close contact under normal circumstances.

(3) The relevant personnel of all participants can only handle one business at a time.

(4) The resource consumption of the processing process is constant, regardless of human and external factors.

(5) The event processing results are satisfactory and can be used for the next step.

(6) Each transshipment node has sufficient capacity to deal with transshipment procedures.

An OOPN-based container export FPMS information flow system model can be constructed in the Petri net environment according to the above modeling process and premise assumptions, and taking into account the functional connection between elements. As shown in Figure 7, the related symbolic meanings are shown in Table 5:

4.6. Detection of FPMS Container Export Information Flow Model Based on Petri Net

4.6.1. Model Reliability Analysis. At present, the most commonly used OOPN model reliability detection method is system deadlock detection. The so-called deadlock refers to a blocking phenomenon caused by competition for resources or due to communication with each other during the execution of two or more processes. If there is no external force, they will not be able to continue to advance. At this time, it is called the system is in a deadlock state or the system has a deadlock. These processes that are always waiting for each other are called deadlock processes [26]. The deadlock here is a dynamic deadlock, which reflects the dynamic characteristics of the Petri net. It can be checked whether there is a deadlock by detecting the reachable centralized deadlock flag of the Petri net. If the deadlock flag is detected in the reachable set, it will exist. Deadlock, otherwise it does not exist [27]. A reachability tree is an effective tool for checking the deadlock identification of Petri nets. It can cover all reachable sets of Petri nets and all deadlocks at the same time. The deadlock detection process of the reachable tree of the object-oriented Petri net can be divided into the following steps:

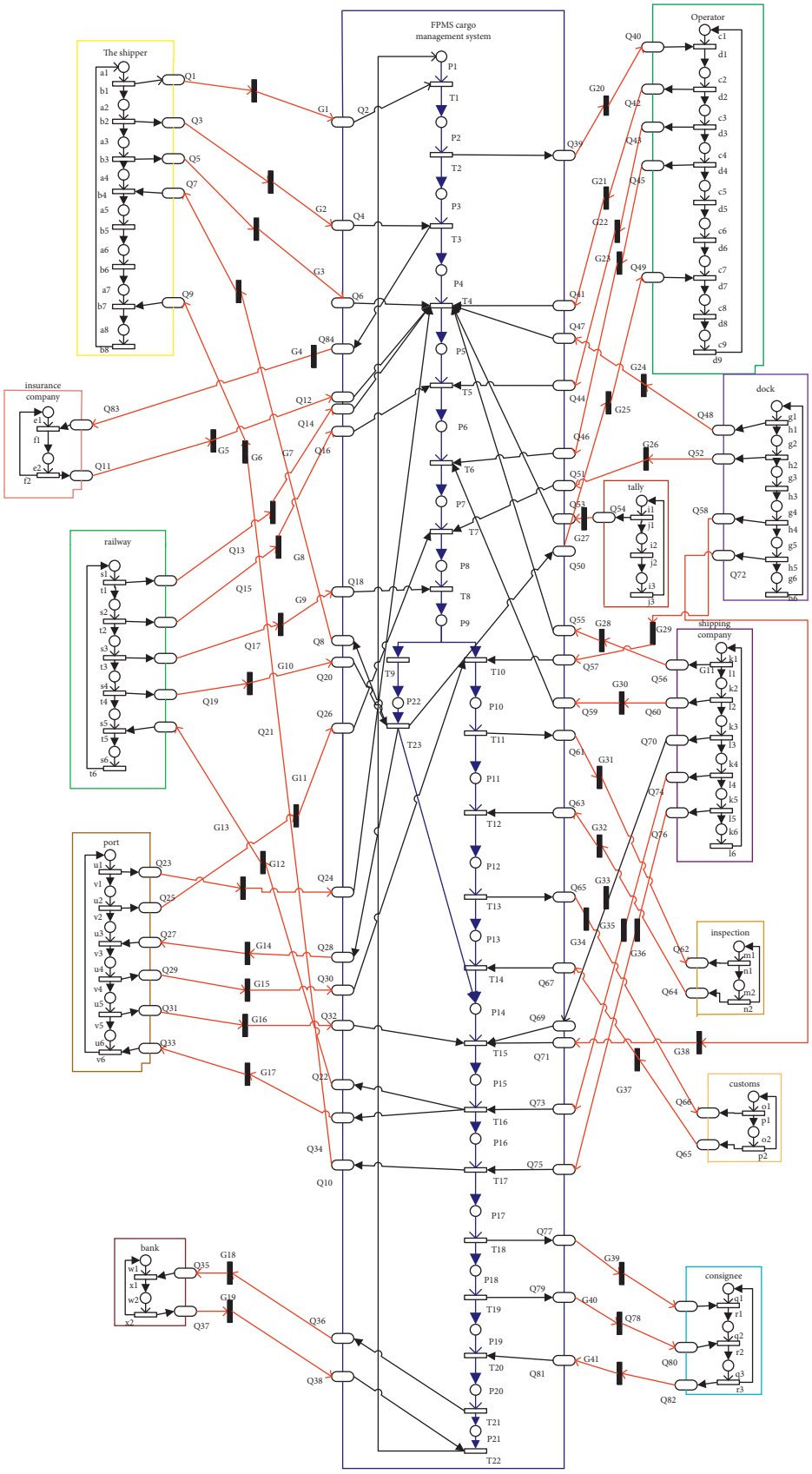


Figure 7: FPMS container export information flow model based on OOPN.

Table 5: FPMS OOPN model symbol meaning of container export information flow.

Symbol	Meaning
P1	Collection of order requirements
P2	The order set
P3	Fill in the request of the whole journey including the waybill
P4	Transportation scheme design
P5	Request for car reservation and suitcase
P6	Booking request
P7	Unloading request
P8	Packing request
P9	Railway transportation and transfer request
P10	Preparation for inspection
P11	Inspecting
P12	Customs preparation
P13	Customs declaration
P14	Export preparation request
P15	Shipment departure request
P16	Maritime transport preparation
P17	Notification of delivery preparation
P18	Issuing bill of lading preparation
P19	Cargo delivery notice
P20	Payment request preparation
P21	Paying
P22	Railway transportation treatment
a1	Transportation demand
a2	Waiting for the whole waybill
a3	Waiting for an agreement
a4	Waiting for transportation preparation
a5	Customs preparation
a6	Preparation for inspection
a7	Sea transport preparation
a8	Cargo delivery notice
c1	Waiting for the shipper to choose
c2	Determining the carrier relationship
c3	Preparation before transportation
c4	Booking, unloading preparation
c5	Customs declaration preparation
c6	Shipping application
c7	Railway transportation preparation
c8	Sea transport preparation
c9	Cargo delivery notice
e1	Waiting for insurance
e2	Signed insurance negotiation
g1	Waiting for the contract to be signed
g2	Receipt of unloading request
g3	Unloading arrangement in progress
g4	Shipping request
g5	Export preparation
g6	Shipping preparation
i1	Waiting for a contract
i2	Tally preparation
i3	Shipping advice
k1	Waiting for a contract
k2	Booking application
k3	Export preparation application
k4	Shipping advice
k5	Sea transportation notice
k6	Delivery notice
m1	Waiting for cargo inspection
m2	In the inspection
o1	Waiting for goods declaration
o2	Customs declaration

Table 5: Continued.

Symbol	Meaning
q1	Waiting for delivery notice
q2	Waiting for bill of lading
q3	Delivery of goods
s1	Waiting for a contract
s2	Booking request
s3	Packing request
s4	Preparation before rail transportation
s5	Shipping advice
s6	Cargo delivery notice
u1	Waiting for a contract
u2	Unloading application
u3	Railway transportation notice
u4	Transfer formalities
u5	Export preparation
u5	Shipment departure preparation
w1	Waiting for payment
w2	Payment preparation
T1	Receive shipping request
T2	Agent selection
T3	Completed
T4	Sign contracts, insurance
T5	The car is ordered and the suitcase is completed
T6	Booking completed
T7	Unloading plan in progress
T8	Packing finished
T9	Railway transportation request
T10	The transfer procedure has been completed
T11	Inspection request
T12	Pass the inspection
T13	Customs request
T14	Customs clearance
T15	Export ready
T16	Shipment departure
T17	In transit by sea
T18	Notice delivery
T19	Issue a bill of lading
T20	Delivery of goods
T21	Paid
T22	Payment completed
T23	In transit by railway
b1	Shipping request
b2	Completed the entire waybill
b3	The agreement was signed
b4	Railway transportation
b5	Customs clearance
b6	Pass the inspection
b7	In transit by sea
b8	Delivery of goods
d1	Selected
d2	Signing a contract, insurance
d3	Booking and picking up
d4	Booking and unloading plan is completed
d5	Customs declaration
d6	Out of the box
d7	In transit by railway
d8	In transit by road
d9	Delivery of goods completed
f1	Signing an insurance claim
f2	Signed insurance completed
h1	Completion of contract

Table 5: Continued.

Symbol	Meaning
h2	Preparation for unloading plan
h3	Unloading list completed
h4	Out of the box
h5	Export configuration list
h6	Shipment, departure
j1	The contract is signed
j2	Check the goods
j3	Shipment departure
l1	The contract is signed
l2	Booking completed
l3	Export preparation
l4	Shipment departure
l5	Sea transport
l6	Delivery of goods
n1	Application for inspection
n2	Pass the inspection
p1	Customs application
p2	Customs clearance
R1	Pick up notice
r2	Receive bill of lading
r3	Receiving finished
t1	The contract is signed
t2	After ordering
t3	Packing completed
t4	Rail transport
t5	Shipment departure
t6	Delivery of goods
v1	The contract is signed
v2	Unloading completed
v3	Railway transportation
v4	The transfer procedure has been completed
v5	Export ready
v6	Shipment departure
x1	Payment request
x2	Payment completed

(1) According to the internal structure of each object, extract its input/output gates and the order of occurrence, and construct an interface equivalent net (IE network).

(2) Combine the different object IE networks in (1) to form the system IE network.

(3) By performing a reachable tree analysis on the established IE network, it is detected whether there is a deadlock. If it exists, the established model needs to be improved to eliminate the deadlock; otherwise, the system is reliable.

Through the analysis of reachable trees and Internet Explorer, it is obvious that there is no deadlock in the freight process management system of FPMS. Similarly, it can be concluded that there is no deadlock in other objects, such as railways, ports, shippers, and through transport operators. Therefore, there is no deadlock in the entire OOPN model.

4.6.2. Bounded Analysis. Boundedness is an indicator to measure whether the resource demand of the system is limited in operation, and it represents the maximum number of tokens that can be obtained during the operation of the system where the library is located. In the running process of the Petri net, there is an integer K that makes the number of tokens in the library in any reachable identifiers of the initial identifiers not exceed K, so that the Petri net is k-bounded.

4.6.3. Performance Comparison. (1) Process analysis: the main flow before and after reconstruction is simulated and compared according to the traditional freight transport flow mentioned in this paper. Before the reconstruction, the freight forwarder's land transportation order was received by the railway, and the freight bill was directly delivered to different ports. All the information is submitted to FPMS, and the order information is sent synchronously to all the intermodal participants after the reconstruction. In the single-node physical host environment, ExSpect software was used to simulate and analyze the Petri net model before and after reconstruction. The simulation diagram is shown in Figures 8 and 9. The main process time was shown in Tables 6 and 7, and the experimental results were shown in Tables 8 and 9.

The simulation results in Table 8 show that (1) the process cycle is long, with an average cycle of about 5528

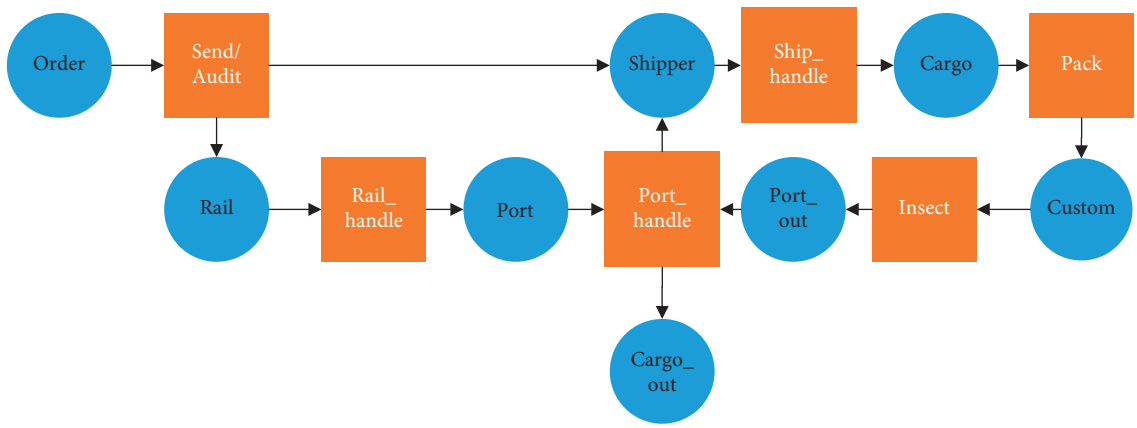


Figure 8: Process simulation screenshot before reconstruction.

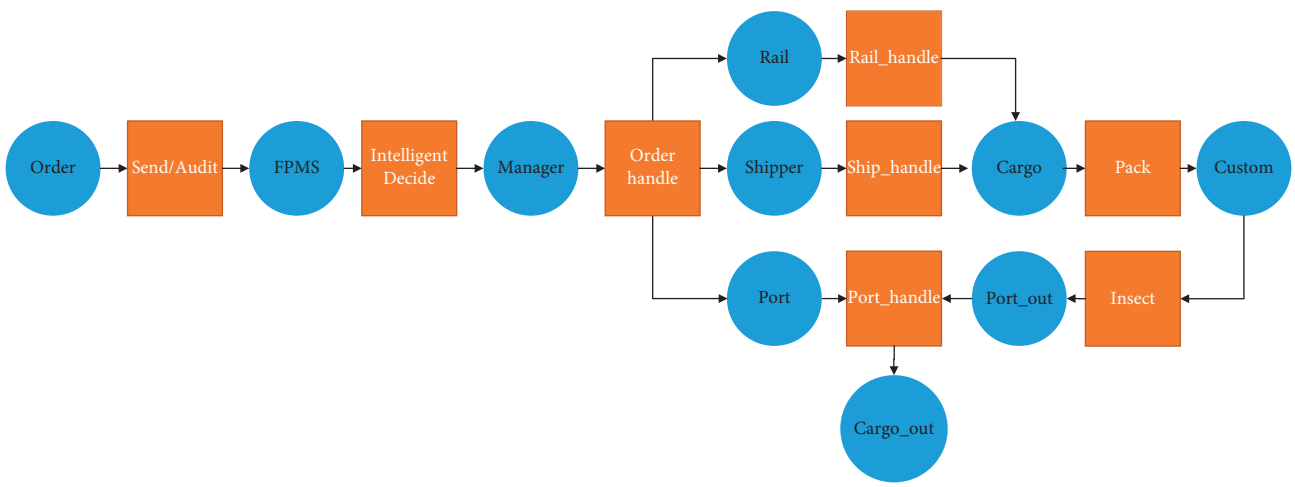


Figure 9: Process simulation screenshot after reconstruction.

Table 6: Time consumption (min) of main business links of the process before reconstruction.

Business	Process	Time
Send/audit	Order delivery and review	1440–2880
Rail_handle	Railway handling	2880–5760
port_handle	Suitcase	2880–5760
Ship_handle	Loading	2880–5760
Pack	Tally	1440–2880
Insect	Customs inspection process	1440–2880

Table 7: Time consumption (min) of main business links of the process after reconstruction.

Business	Process	Time
intelligentDecide	Order collection and intelligent matching, order submission to the freight process management system, the system network for real-time transmission of messages	2–3
Order_handle	The cloud platform processes/distributes orders	2–3
Rail_handle	Railway handling	2880–5760
Port_handle	Suitcase	2880–5760
Ship_handle	Loading	2880–5760
Pack	Tally	1440–2880
Insect	Customs inspection process	1440–2880

Table 8: Simulation results of freight flow before reconstruction.

Group number	Number of process samples	Average process cycle time (min)	The variance
1	83	5631.1	2310.1
2	127	5579.8	1924.3
3	130	5713.4	1894.6
4	122	5418.3	1806.8
5	119	5297.4	1993.5

Table 9: Simulation results of freight flow after reconstruction.

Group number	Number of process samples	Average process cycle time (min)	The variance
1	75	3981.7	234.4
2	140	3511.8	199.8
3	94	3420.4	213.3
4	133	3313.1	220.4
5	129	3403.6	210.7

minutes, and (2) large variance value, large dispersion degree, process instability, and uncertainty.

The analysis of the process shows that (1) the process has a long cycle because the serial structure is adopted, and the information transfer process needs to consume time. Meanwhile, the process is processed in advance according to experience. (2) The process has a large dispersion degree, which indicates that the process is greatly influenced by uncertain factors, and the system stability is weak.

The simulation results in Table 9 show that (1) the cycle time of this process is shorter than that of the original process, with an average cycle time of about 3526 minutes. Compared with the original process after optimization, the average cycle time of the freight flow is shortened by 36.21%. (2) The variance value is smaller, the system dispersion degree is reduced, and the process stability is improved.

According to the analysis of the process, the reconstructed freight flow management system simplifies the information interaction and processing process in the freight process, reduces the repeated input of data, and thus shortens the time of information handover. In addition, the business coherence and parallelism can be improved because of the real-time order information, and the freight transport efficiency can be improved significantly. After the electronic standardization of the system, the uncertainty is reduced, the process stability is greatly improved, and the process synchronization and efficiency of operation are realized. The system load is relatively balanced by adopting the Big Data platform processing mode in the cloud environment, while the data processing capacity is greatly improved, the resource utilization rate is higher, and the execution rate is also higher.

5. Conclusion

Based on the hindrance of the overall development of railway-river business at present, combined with the current situation of the decentralized layout of the intermodal transportation information system, this paper puts forward the idea of constructing the information layout of the cloud platform for intermodal transportation, and takes Chinese container as an example to realize the optimization of the freight business process of the joint transport of railway-river and freight. The application of Big Data, cloud computing and other technologies provides a large amount of real-time sharing, connectivity and interaction information for intermodal transportation operators, railways, ports, and shipping companies, saves the time cost of business development, and improves the business efficiency and security; for government agencies, such as customs clearance and inspections, the unified resource management mode provides the possibility of real-time query of all orders to ensure the quality of each cargo. There are records to track, which improves the administrative efficiency; for the container truck distribution company and insurance bank, it provides real-time interactive application information, which is conducive to timely adjust the service mode and better adapt to the business needs. In a word, it is very necessary to adopt the "big centralized" information management mode under the Big Data platform to realize the business process reconstruction of intermodal transportation to meet the business development needs of current intermodal transportation informatization.

Due to the large number of participating institutions of the intermodal information platform, the large business complexity, and the large number of legacy systems, there is no precedent for the introduction of cloud computing technology into intermodal information construction and the potential knowledge mining of intermodal Big Data. There is still a broad development space in the construction of the intermodal cloud platform, and long-term practice and exploration are still needed. This article has the following points to focus on in the follow-up.

5.1. Standardization of Cloud Platform for Rail Water Transport. Due to the imperfect information standards of rail water intermodal transportation in China, most of the current application architectures cannot adapt to the cloud environment. Although the intermodal transportation cloud platform is environmentally compatible with the legacy applications, a lot of reconstruction work

is still needed for the current information. Due to the different business processes of different ports and the different information exchange standards, it is difficult to implement the application and information integration process on the intermodal cloud platform, which also requires a lot of human and material resources. Although the simple migration can realize the unified management of the application, it cannot effectively use the virtual resources to optimize the cloud service mode.

5.2. Information Isolation between Intermodal Applications. A large number of intermodal applications reside in the cloud and share the virtual computing resources and storage resources of the cloud in the intermodal cloud platform environment. How to isolate the internal and external service data while ensuring the information sharing requirements and protect the privacy of the business application information of the intermodal departments involved are also issues that need to be considered. Although virtualization technology provides a data isolation mechanism, more in-depth research is needed on data partitioning, protection, and utilization.

5.3. Data Fusion in the Internet-of-Things Environment. With the advent of the Internet-of-Things era, the cross-border integration of the interoperability industry and the Internet of Things is the only way to improve business and management efficiency, realize intelligent management, and maintain advancement. In the Internet-of-Things environment, massive sensor data will enter the intermodal cloud platform. How to use the powerful data processing capabilities of cloud computing to realize the analysis and processing of massive real-time sensor information will also be a severe challenge.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Furong Jia and Lin Sun contributed to the work equally and should be regarded as co-first authors.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (grant nos. 2017YFB1200702 and 2016YFC0802208), National Natural Science Foundation of China (project no. 61703351), Sichuan Science and Technology Program (project nos. 2018RZ0078 and 2019JDR0211), Science and Technology Plan of China Railway Corporation (project no. P2018T001), Chengdu Soft Science Research Project (grant nos. 2017-RK00-00028-ZF and 2017-RK00-00378-ZF), and the Fundamental Research Funds for the Central Universities (grant nos. 2682017CX022 and 2682017CX018. This research was also funded by Sichuan Agricultural University education reform projects X2013039 and X2014025, "Agricultural Information Engineering," Sichuan key laboratory of higher education.

References

[1] Y. Tingyu, P. Xiaoqian, C. Dingjun et al., "Research on trans-region integrated traffic emergency dispatching technology based on multi-agent," *Journal of Intelligent and Fuzzy Systems*, vol. 38, pp. 1–12, 2020.

[2] S. Liu, C. Yin, D. Chen et al., "Cascading failure in multiple critical infrastructure interdependent networks of syncretic railway system," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[3] D. Chen, S. Ni, C. A. Xu, and X. Jiang, "Optimizing the draft passenger train timetable based on node importance in a railway network," *Transportation Letters*, vol. 11, no. 1, pp. 20–32, 2019.

[4] K. Che, X. Wang, Possible sustainability of intermodal transportation in Africa," *Open Journal of Applied Sciences*, vol. 9, no. 4, pp. 139–158, 2019.

[5] Z. Li, "Design and implementation of credit information system based on data centralization mode," *M. S. thesis, Department of Computer Applications Technology, Hunan University, Changsha, China*, 2010.

[6] P. He and W. Li, "Modeling and analysis of logistics distribution process based on Stochastic Petri net," *Journal of Wuhan University of Technology*, vol. 32, no. 3, pp. 434–436, 2010.

[7] F. Cao, "New ideas on regional economic development and industrial transformation under "jinjiang experience" A new logistics model based on the establishment of three-port linkage," *Advances in Economics, Business and Management Research*, vol. 71, pp. 86–94, 2018.

[8] X. Chen, "An empirical analysis of competitiveness of yangshan deep-water port: the application of intermodal transportation and automated container terminal," in *Proceedings of the 2019 2nd International Workshop on Advances in Social Sciences (IWASS 2019)*, pp. 111–122, London, UK, November 2019.

[9] T. Chen, Q. Huang, and S. Ni, "Research on the architecture scheme of the information platform of railway-river combined transport based on cloud computing," *Logistics technology*, vol. 23, pp. 404–407, 2014.

[10] Meng, "The research based on private cloud computing information interaction model," in *Proceedings of the 2015 International Conference on Automation,Mechanical Control and Computational Engineering (AMCCE 2015)*, Ji'nan, China, April 2015.

[11] S. Cao, S. Chu, and B. Zhang, "Practice and research on construction management of engineering project for new-generation large-scale high-speed railway passenger station," in *Proceedings of the 8th International Symposium on Project Management, China (ISPM2020)*, pp. 378–396, Aussino Academic Publishing House, Beijing, China, July 2020.

[12] H. W. Wang, C. H. Chen, D. Y. Cheng et al., "A real-time pothole detection approach for intelligent transportation system," *Mathematical Problems in Engineering*, vol. 2015, Article ID 869627, 7 pages, 2015.

[13] W. Jansen and T. S. P. Grance, *Guidelines on Security and Privacy in Public Cloud Computing*, National Institute of Standards & Technology, Gaithersburg, MA, USA, 2011.
[14] C. L. Iacovou, I. Benbasat, and A. S. Dexter, "Electronic data Interchange and small organizations: adoption and impact of technology," *MIS Quarterly*, vol. 19, no. 4, pp. 465–485, 1995.
[15] J. Mantas, A. Hasman, and H. Arie, *Electronic Data Interchange*, IOS Press, Amsterdam, Netherlands, 2002.
[16] P. García-Sánchez, J. González, P. A. Castillo, M. G. Arenas, and J. J. Merelo-Guervós, "Service oriented evolutionary algorithms," *Soft Computing*, vol. 17, no. 6, pp. 1059–1075, 2013.
[17] R. Singh Bhadoria, N. S. Chaudhari, and G. Singh Tomar, "The Performance Metric for Enterprise Service Bus (ESB) in SOA system: theoretical underpinnings and empirical illustrations for information processing," *Information Systems*, vol. 65, 2017.
[18] B. Benatallah and H. R. Motahari Nezhad, "Service oriented architecture: overview and directions," in *Advances in Software Engineering*, E. Börger and A. Cisternino, Eds., vol. 5316, 2008.
[19] Sonic Software Corporation, *Sonic ESB: An Architecture and Lifecycle Definition*, Sonic White Paper, Novato, CA, USA, 2005.
[20] S. Hudson, "The Enterprise service bus: disruptive technology for software infrastructure solutions," *IDC Insight*, vol. 1, 2003.
[21] E. Kozan, "Optimising container transfers at multimodal terminals," *Mathematical and Computer Modelling*, vol. 31, no. 10-12, pp. 235–243, 2000.
[22] R. Rajkovic, N. Zrnic, S. Kirin, and B. Dragovic, "A review of multi-objective optimization of container flow using sea and land legs together," *FME Transaction*, vol. 44, no. 2, pp. 204–211, 2016.
[23] C. WANG, D. César, and W. WANG, "Port integration in China: temporal pathways, spatial patterns and dynamics," *Chinese Geographical Science*, vol. 25, no. 5, pp. 612–628, 2015.
[24] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.
[25] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
[26] W. Nie, *Business Process Optimization of Railway Logistics Center Based on Petri Net*, Southwest Jiaotong University, Chengdu, China, 2012.
[27] L. Huang, *Research on Multithreaded Deadlock Detection Based on Petri Nets*, University of Science and Technology of China, Hefei, China, 2015.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5546127, 17 pages
https://doi.org/10.1155/2021/5546127

Research Article

Extraction of Optimal Measurements for Drowsy Driving Detection considering Driver Fingerprinting Differences

Yifan Sun, Chaozhong Wu, Hui Zhang, Yijun Zhang, Shaopeng Li, and Hongxia Feng

Intelligent Transportation Systems Research Center, Wuhan University of Technology, 1178 Hepingdadao Street, Wuchang District, Wuhan 430063, Hubei, China

Correspondence should be addressed to Hui Zhang; zhanghuiits@whut.edu.cn

Received 21 January 2021; Revised 14 June 2021; Accepted 11 August 2021; Published 31 August 2021

Academic Editor: Chi-Hua Chen

Contributions of measurements for detecting drowsy driving are determined by calculation parameters, which are directly related to the accuracy of drowsiness detection. The previous studies utilized the same Unified Calculation Parameters (UCPs) to compute each driver's measurements. However, since each driver has unique driving behavior characteristics, namely, driver fingerprinting, Individual Drivers' Best Calculation Parameters (IDBCPs) making measurements more discriminative for drowsiness are various. Regardless of the difference in driver fingerprinting among the drivers being tested, using UCPs instead of IDBCPs to compute measurements will limit the drowsiness-detection performance of the measurements and reduce drowsiness-detection accuracies at the individual driver level. Thus, this paper proposed a model to optimize calculation parameters of individual driver's measurements and to extract individual driver's measurements that effectively distinguish drowsy driving. Through real vehicle experiments, we collected naturalistic driving data and subjective drowsy levels evaluated by the Karolinska Sleepiness Scale. Eight nonintrusive drowsiness-related measurements were calculated by double-layer sliding time windows. In the proposed model, we firstly applied the Wilcoxon test to analyze differences between measurements of the awake state and drowsy state, and constructed the fitness function reflecting the relationship between the calculation parameters and measurement's drowsiness-detection performance. Secondly, the genetic algorithms were used to optimize fitness functions to obtain measured IDBCPs. Finally, we selected measurements calculated by IDBCPs that can distinguish drowsy driving to constitute individual drivers' optimal drowsiness-detection measurement set. To verify the advantages of IDBCPs, the measurements calculated by UCPs and IDBCPs were, respectively, used to build driver-specific drowsiness-detection models: DF_U and DF_I based on the Fisher discriminant algorithm. The mean drowsiness-detection accuracies of DF_U and DF_I were, respectively, 85.25% and 91.06%. It indicated that IDBCPs could enhance measurements' drowsiness-detection performance and improve the drowsiness-detection accuracies. This paper contributed to the establishment of personalized drowsiness-detection models considering driver fingerprinting differences.

1. Introduction

Drowsy driving is a typically dangerous driving behavior, which seriously threatens traffic safety and causes substantial financial costs to individuals and society [1–3]. Drowsy driving is a common phenomenon that results from multiple underlying reasons, including prolonged driving hours and sleep deprivation [3, 4]. A review of previous studies indicated that there was an obvious association between drowsy driving and the risk of traffic accidents [4, 5]. The studies pointed out that 7% of all accidents and 16.5% of all road casualties were related to drowsy driving [6]. And, in the European Union, about 20% of commercial transport crashes were caused by drowsy drivers [7]. It was found that the risk of a near-crash event was significantly increased when drivers were drowsy after a night shift [8]. Besides, it is a problem to judge whether a traffic accident is caused by drowsiness [9]. Unlike drunk driving, many accidents related to drowsiness have not been reported due to the lack of objective criteria for judging drowsiness occurrence, and the actual hazards of drowsiness driving may be more serious [10].

Therefore, it is of great significance to study the accurate real-time anti-drowsiness warning system.

Drowsiness-detection models are at the core of the anti-drowsiness warning systems, which are mainly divided into two categories: intrusive and nonintrusive [11, 12]. The intrusive drowsiness-detection models rely on contact devices to collect physiological data, such as electroencephalograms [13], electromyography [14], etc. For example, researchers [15] obtained the signal of heart rate variability (HRV) from surface electrocardiogram and trained drowsiness detector using 7 features derived from HRV, and the positive predictive value reached 0.96. Although the drowsiness-detection using physiological signal is more accurate, the intrusiveness of collecting data limits the practicability [14]. The nonintrusive drowsiness-detection models use noncontact sensors to collect data nonintrusively to detect drowsy driving, such as steering wheel angle, lane position, eye movement data, etc. [14, 16]. The nonintrusive drowsiness-detection models are more practical because they have little interference with drivers and fewer restrictions [14]. So, nonintrusive drowsiness-detection methods have always been a hotspot in drowsy driving research [12, 17]. As reported in much literature, the researchers established various drowsiness-detection models based on nonintrusive measurements, such as lane position derived measurements [18], steering wheel angle derived measurements [17], and the ocular movement derived measurements [19] and obtained relatively accurate drowsiness-detection results. Thus, this paper concentrated on nonintrusive measurements and models of drowsiness detection.

The sensitivity of nonintrusive measurements to drowsiness depends on the time window setting method and calculation parameters of measurements, which greatly determines the accuracy of drowsiness-detection [18, 19]. Many studies concerning measurements and detection methods of drowsy driving have been conducted, which can be found in some review literature [11, 12, 14, 20–22]. This paper focuses on the optimization of calculation parameters of nonintrusive drowsiness measurements. Therefore, we summarize some literature that typically uses nonintrusive measurements to detect drowsiness and gives specific computational parameters. Table 1 is about the apparatus, time window setting, and parameters for nonintrusive measurements calculation in partial drowsy driving studies. It can be seen that time window setting and calculation parameters of nonintrusive measurements in studies are inconsistent, which indicates that these factors affect research outcomes of drowsy driving. Therefore, the calculation parameters of the measurements need to be optimized to improve the drowsiness-detection performance of nonintrusive measurements.

Most of the drowsy driving studies calculated the measurements of each driver by the same Unified Calculation Parameters (UCPs), without considering individual differences in measurements' calculation parameters, which are results of driver fingerprinting differences among drivers. It has been discovered that each person not only has unique physical characteristics but also behavioral characteristics [31]. For example, every person has his or her behavioral characteristics in shopping, writing, reading, etc. Driver fingerprinting is each driver's unique driving behavior characteristics during the driving duration [32], which are reflected in the steering wheel angle, speed, eye movement, and other driving behavior data [33, 34]. The driver fingerprinting is stable and unique [33–35]. So, several researchers have tried to describe driver fingerprinting to identify the drivers. Finker et al. [35] collected multiple CAN-bus signals by field experiments, and cepstral feature extraction techniques were used to analyze brake pedal signals and gas pedal signals. They used artificial neural networks to learn driver fingerprinting characteristics and performed driver-identification. The identification accuracy yielded 84.6%. Xun et al. [36] used actual vehicles to collect naturalistic driving behavior data, including speed, steering wheel angle, accelerator pedal signal, etc. They built a model to achieve driver fingerprinting based on a convolutional neural network and support vector domain description and used driver fingerprinting to accurately identify drivers. The above studies utilized the differences in driver fingerprinting among drivers to accurately identify drivers, which proves that drivers have stable driving behavior characteristics that are different from others. And, the differences in driver fingerprinting were also the reason for the individual differences in some aspects of drowsy driving studies, such as the calculation parameters, data distribution, and drowsiness-detection threshold of drowsiness-related measurements [37–39].

Many scholars have noticed individual differences caused by driver fingerprinting differences in drowsy driving detection research [18, 38, 40, 41]. Most drowsiness-detection models were generalized models that were trained by blending measurements of all drivers, and these models detected the drowsiness of each driver utilizing the same drowsiness threshold [14]. The differences in driver fingerprinting were the important reasons for the low reliability of the nonintrusive generalized drowsiness-detection models [39, 42, 43]. It was frequently mentioned that, due to the driver fingerprinting differences, drivers differed considerably in data distribution of measurements such as SDLP [39], the standard deviation of steering wheel movement (SDSWM) [38], PERCLOS [39], eye closure duration [43], etc. For instance, Inger et al. [37] analyzed the correlation between self-reported drowsiness level and blink duration and SDLP. The results indicated that there were obvious differences among drivers in the distribution of blink duration and SDLP under various levels of drowsiness. They also pointed out that the drowsiness-detection at the individual driver level suffered from systematic errors if the drowsy driving was detected by the same drowsiness-detection threshold for all drivers. Moreover, researchers explored reasons for differences in driver fingerprinting that affected the drowsiness-identification. Silveira, et al. adopted features derived from electrocardiographs and eye movement signals to build the classifiers of subject-dependent driver sleepiness. They found that physiological signals presented obvious individual characteristics, while subject-independent drowsiness classifiers that ignored individual differences performed worse [44]. Similarly, Persson, et al. used HRV measurements to build a drowsiness classifier based on machine learning and pointed out that the accuracy

Table 1: Summary of information in drowsy driving studies.

References	Apparatus	Time window setting	Nonintrusive measurements	Calculation parameters
Li et al. [23]	RV	SSTW	ApEn	Time window is 60 s
Wakita et al. [24]	DS	SPTW	VCF and DCF	Time window is 0.6 s
Rossi et al. [25]	DS	SPTW	Mean and standard deviation of steering error and SDLP	The time window is 60 s
Henni et al. [26]	RV	SPTW	Eye closure duration, mean of speed, and steering wheel movement	The time window is 30 s
Feng et al. [27]	DS	SPTW	PERCLOS, average eye-opening level, PNS, and SDSWA	The time window is 30 s
McDonald et al. [28]	DS	SPTW	Steering wheel angle	The time window is 60 s
Zandi et al. [19]	DS	SSTW	PERCLOS, saccade rate, and blinking rate	The time window ranges from 30 s to 60 s
Cheng et al. [29]	DS	SPTW	PERCLOS blinking duration, and blinking rate	Time window is 60 s
Zhang et al. [30]	RV	DPTW	SDSWA and PNS	The first-layer time window is 60 s, the second-layer time window is 15 s

RV = real vehicle, DS = driving simulator, SSTW = single-layer sliding time window, SPTW = single-layer parallel time window, DPTW = double-layer parallel time window, ApEn = approximate entropy from steering wheel angle, VCF = velocity change in the car following scene, DCF = distance change from the vehicle in front, SDLP = standard deviation of lane position, PERCLOS = percentage of eyelid closure, SDSWA = standard deviation of steering wheel angle, and PNS = percentage of nonsteering.

decreased dramatically for a new driver because of individual differences [45]. Yan et al. [41] analyzed driver fingerprinting differences in drowsiness-detection models and found that differences in measurement distribution among drivers could be beyond differences caused by drowsiness when all drivers' measurements were mixed to train models, which decreased the correlation between measurements and drowsiness. Therefore, the accuracies of drowsiness-detection models without driver fingerprinting differences were impaired.

For eliminating the negative influence of driver fingerprinting differences on drowsiness-detection accuracy, scholars attempted to use measurements of individual drivers to establish driver-specific models and achieve more accurate drowsiness-detection [39, 41, 43, 46, 47]. Chu et al. [47] recorded naturalistic driving data by actual vehicles, and trained drowsiness-detection models by the speed and lane departure value of individual drivers, based on the combined model of neural network and support vector machine. The drowsiness-identification rate of the model for some drivers exceeded 90%, which verified that driver-specific models could improve the accuracy of drowsiness-detection. You, et al. [46] handled differences in eye movement behavior data among drivers, extracted ocular measurements like eyes aspect ratio, and used ocular measurements of individual drivers to establish the driver-specific drowsiness-detection models. The accuracy of the proposed model reached 94.8%, which outperformed generalized drowsiness-detection models. Wang et al. [39] collected multisource driving behavior data by simulating driving experiments and utilized 23 nonintrusive measurements of individual drivers to build the personalized drowsy driving models based on the multilevel logit model. The results verified that the drowsiness-detection accuracy of the personalized model considering driver fingerprinting differences was higher. Naurois et al. [48] used data of a group of drivers to train the drowsiness-detection model based on artificial neural networks and used adaptive learning to personalize the drowsiness-detection model for a new driver. The model enhanced drowsiness-detection performance by roughly 40%.

The above driver-specific drowsiness-detection models improved the accuracy of drowsiness-detection, which was attributed to adopting individual drivers' specific drowsiness-detection thresholds. However, in the above driver-specific models, measurements of each driver were still computed by UCPs rather than Individual Drivers' Best Calculation Parameters (IDBCPs), which restrained contributions of measurements for detecting drowsiness. Due to differences in driver fingerprinting among drivers, IDBCPs that made measurements more discriminative for drowsiness were various [18, 49]. For some drivers, IDBCPs of measurement are very different from UCPs. And, using UCPs to compute individual driver's measurements restrains the drowsiness-detection performance of measurements, which decreases the accuracy of driver-specific drowsiness-detection models. Similarly, Zhang et al. [18] found that there were considerable differences in IDBCPs and using the UCPs to calculate measurements might weaken the correlation between measurements and drowsiness levels for some drivers. Therefore, the purpose of this paper was to build a novel model to optimize the measurements' calculation parameters of the individual driver. And, measurements calculated by IDBCPs which could validly distinguish drowsiness were chosen to compose individual drivers' optimal measurement set of drowsy driving. This paper contributes to drowsiness measurement calculations and the establishment of drowsiness-detection models considering driver fingerprinting differences.

The rest of this paper is arranged as follows. In Materials and Methodology, we described the experiment details, measurement calculation method, the extraction model of optimal drowsy driving measurements of individual drivers, and the validation method of the drowsiness-detection advantages of measurements calculated by IDBCP. In Results, we presented the driver fingerprinting differences of measurements using UCPs, the results of the model, and the drowsiness-detection advantages of using IDBCPs to calculate nonintrusive measurements. The discussion of the results, the important contributions, and the application prospects of this paper are introduced in Discussion. And, the summary of research work, research limitations, and future work are given in Conclusions.

2. Materials and Methodology

2.1. Experimental Design

2.1.1. Participants. According to the actual proportion of male and female commercial vehicle drivers locally, 35 male drivers and 5 female drivers were recruited as participants and numbered. All participants were professional drivers with rich driving experience and proficient driving skills. Their age ranged from 34 years old to 57 years old (mean = 46.83, SD = 5.62); the driving age ranged from 3 years to 32 years (mean = 16.53, SD = 6.10). Participants did not have any sleep-related disorders and other physical or mental illnesses, and they took no pharmaceuticals within 1 month before the experiments. On the co-pilot, we arranged one coach with 30 years of driving experience as a safety officer to accompany every participant. The safety officer took countermeasures during emergencies and was trained to inquire and record the participants' self-reported drowsiness level using the Karolinska Sleepiness Scale (KSS). The KSS ranges from one to nine, where one and nine, respectively, represent extremely alert and extremely drowsy [50]. The safety officer has a rich long-distance driving experience and strong anti-fatigue abilities. To prevent the safety officer from using KSS due to excessive fatigue, the safety officer should have a good rest when arriving at the service area. Besides, during driving duration, the safety officer kept relatively alert by drinking coffee and refreshing energy beverages and listening to pop music, etc.

2.1.2. Devices. The major experimental devices and routes are shown in Figure 1. The experimental vehicle was modified from the actual vehicle, equipped with a variety of sensors. The multisource naturalistic driving behavior data were collected timely. The three-way HD camera (frame rate was 30 Hz) could capture the driver's facial characteristics, driving operation details, and the traffic environment outside the vehicle including road linearity and traffic flow. The HD cameras were Logitech C910. The driver state sensor (sampling frequency was 60 Hz) was W5Y-DSSV3 (AR-ES5430SM) produced by Seeing Machines, which used an image recognition algorithm to monitor the driver's eyes and eyelid movements. The Mobileye C2-270 (sampling frequency was 15 Hz) could detect lane lines and output the distance between the vehicle center and the lane lines. The inertial navigation system (sampling frequency was more than 20 Hz) could collect acceleration, velocity, and GPS, which was RT2500 developed by Oxford Technical Solutions. The industrial computer could store raw multiple-source experimental data, which was SIMATIC Rack PC 847B of SIEMENS. The steering wheel angle sensor was LWS 5 of BOSCH (sampling frequency was more than 20 Hz), which collected the steering wheel angle. Besides, other materials were also used, such as the KSS, demographic information scale, etc.

2.1.3. Implementation Process. Before experiments, the participants were notified of the important information including the experimental process and the possible risk, and each participant signed informed consent. Participants were trained to understand KSS correctly for ensuring the reliability of self-reported drowsiness levels. Then, participants were familiarized with the experimental vehicles and made sure not to take beverages, such as caffeine, that would affect the measure of drowsiness. On the night before the experiment, participants kept a normal schedule and ensured adequate sleep duration. Before driving, participants were required to report their sleep duration the previous night and current KSS.

In the experiment, participants were required to drive the car along the route shown in Figure 1 with their usual driving habits. The experimental road section was G70 (two-way four-lane) Expressway from Wuhan to Xiangyang, China. The road section was flat and relatively straight. Its traffic flow was simple. So the influence of road shape and other vehicles' disturbance on driving behavior could be eliminated such as steering and lane-keeping, which was out of the scope of the current work. The factor of weather, such as snow, rain, and fog, can also affect naturalistic driving behavior [51], which can interfere with the analysis of the effects of drowsiness on driving behaviors. To control irrelevant variables, all experiments were carried out under the condition of well-lit weather to ensure that the temperature, humidity, and other environmental factors were consistent. Besides, each participant started the experiment at 9 Am on weekdays to avoid the influence of traffic volume changes. All participants entered the G70 highway from the Fuhe toll station. After about 2 hours, participants reached the Suizhou service area and took a 1-hour break. Then, participants drove to Xiangyang north toll station, turned around at this point, and returned to the start point. The total driving time was about 6 hours and the total travel was about 600 kilometers in this experiment. During the process of driving, the noise level was controlled to keep a comfortable environment and prevent noise from influencing the physiological and psychological state of participants. Participants were required to drive at a speed limit of 120 km/h. The same safety officer inquired to record 40 participants' self-report KSS every 5 minutes.

After completing the whole experiment, participants were paid for their participation. And, the experimental designs met the guidelines of the local ethics committee and the

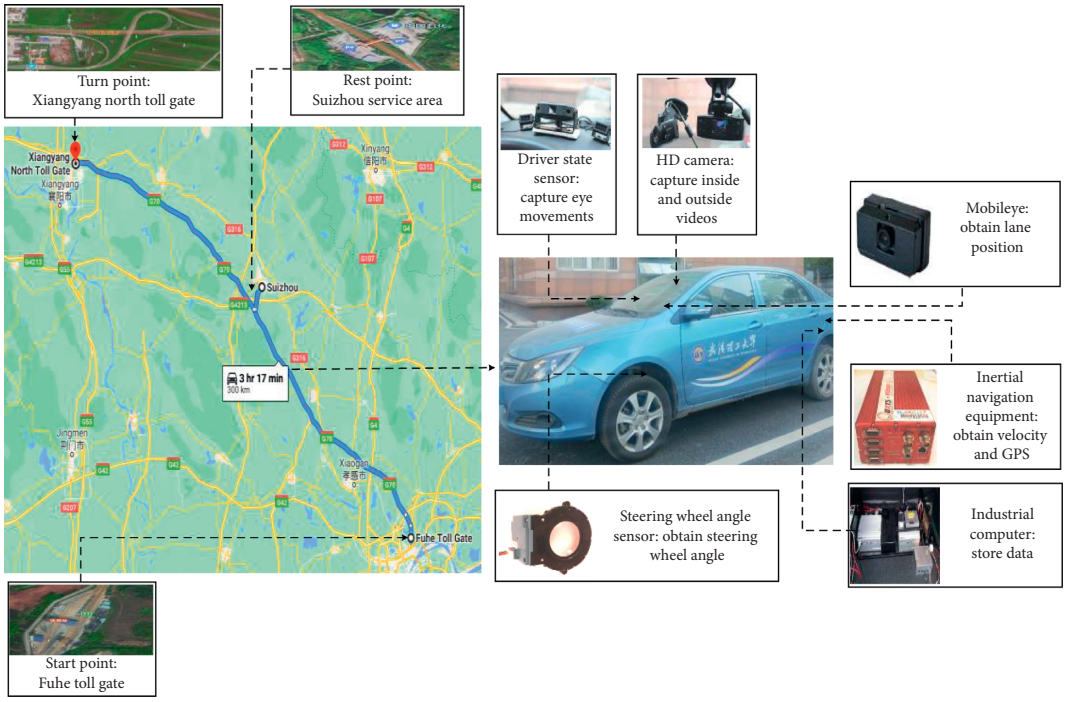


Figure 1: Diagram of experiment apparatuses and routes.

personal information of participants was properly secured. Finally, 35 participants' qualified experimental data were obtained. Valid data of 5 participants were not collected due to poor experimental conditions or equipment problems.

2.2. Data Preprocessing. According to the flow in Figure 2, raw experimental data were preprocessed to obtain the drowsy driving measurements that meet the requirements of the model.

2.2.1. Synchronizing Multisource Data. Based on the timestamp recorded by the sensor, we synchronized multisource data including videos, self-reported KSS, lane position, steering wheel angle, velocity, and so on. Because the driver's drowsiness level was evaluated every 5 minutes, we divided the raw data into samples within 5 minutes to avoid the sample crossing two KSS.

2.2.2. Choosing Data of the Continuous Driving Scene. The driving behavior in the continuous driving scene on the motorway was chosen to analyze. The driving operation is more monotonous when the driver is driving continuously on the motorway, and the risks of drowsy driving are greater because of the high speed [39]. According to the videos outside the vehicle, we extracted the driving duration under the continuous driving scene and selected the data of the continuous driving scene.

2.2.3. Setting Double-Layer Sliding Time Window and Calculating Measurements. The research in the field experiments found that the drowsy driving state generally lasted 15–75 seconds, while the typical drowsy operation characteristics duration was generally 5–20 seconds [30]. When measurements are calculated by data in the entire drowsy state duration, the calculating average effect covers the drowsy operating characteristics [1, 18]. Thus, the double-layer sliding time window was proposed to divide raw data into samples and calculate measurements.

Firstly, we set the first-layer sliding step (S_1) and time window (T_1) within every 5 minutes. The experimental data in T_1 were data of a sample, and T_1 was a fundamental unit for calculating measurements and detecting drowsiness. The data in one T_1 could be used to calculate one measurement value, which was one measurement sample. Secondly, within each T_1 time window, we set the second-layer sliding step (S_2) and time window (T_2). The data in T_2 were used to calculate the measurement. To magnify the drowsy driving characteristics in the sample (T_1), the maximum measurement for all T_2 in every T_1 was chosen as the final measurement value of the sample (T_1). For instance, when calculating SDLP, if $T_1 = 60$ s, $S_2 = 5$ s, $T_2 = 20$ s, then T_1 included nine T_2 and the lane position data in each T_2 can be used to calculate one SDLP; we chose the maximum SDLP of nine T_2 as the final SDLP of the sample T_1. The number of T_2 was determined by the size of S_2. The smaller the S_2 is, the more T_2 can be obtained to select max drowsiness measurement value within T_1. But if the amount of T_2 was too large, the calculation amount of the model would increase and the calculation speed of the model would reduce. Besides, there are certain gaps between typical fatigue operating features, S_2 cannot be too small. Thus, for reducing the amount of computation, avoiding or missing the best S_2, and reducing the degree of freedom, we design the optimization range of S_2 to be an integer between 2 and 5.

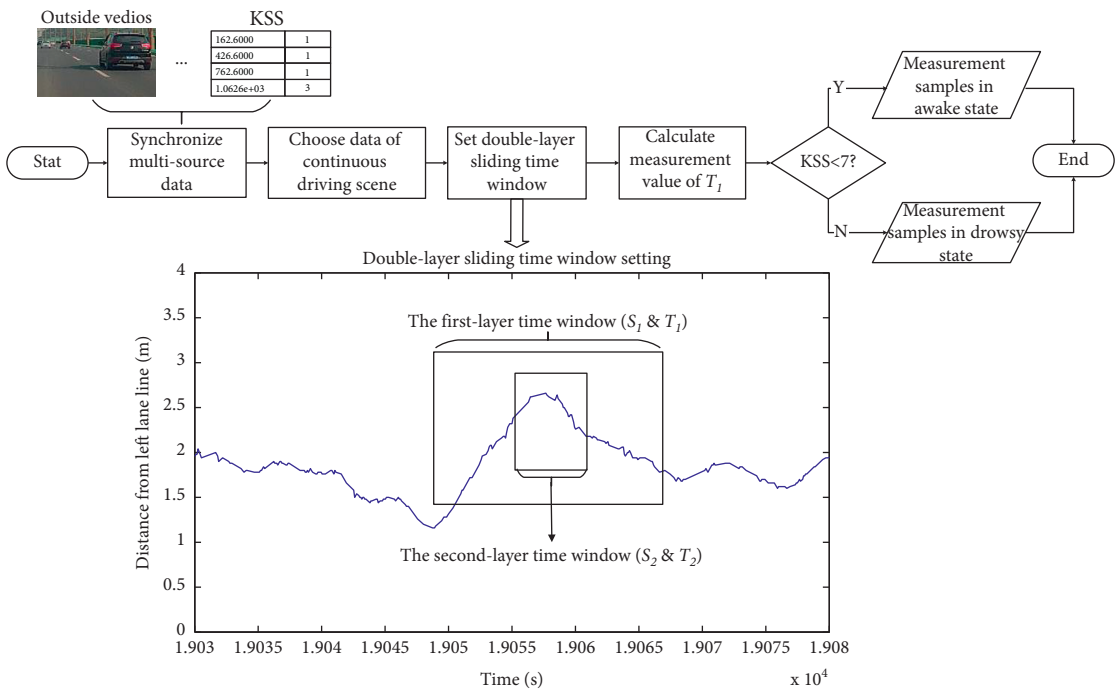


Figure 2: The workflow of data preprocessing.

2.2.4. Dividing Measurement Samples into Samples of Awake State and That of Drowsy State. The KSS of T_1 was defined as the KSS of measurement samples. Referring to the literature [18, 52], measurement samples with KSS less than 7 belonged to samples of the awake driving state; otherwise, they belonged to samples of the drowsy driving state. In Figure 3, we drew the KSS overtime of 35 participants. The KSS equals zero means stop because of rest or some breakdowns. As shown in Figure 3, with the increase in driving time, KSS gradually increased. Before and after rest in the service area, the KSS peak appeared with the increase of driving time. And after the rest in the service area, KSS decreased to a certain extent, but KSS rapidly increased with the increase of driving time.

2.3. Measurement Calculation. Drowsiness can affect the driver's steering wheel rotation, eye movement, and other aspects [12]. According to related literature [12, 18, 39], eight drowsiness-related nonintrusive measurements of different categories were chosen to analyze. In the previous studies, measurements of all drivers were calculated by UCPs. For comparing with the drowsiness-detection contributions of measurements calculated by IDBCPs, we also used suitable UCPs to calculate measurements.

We utilized the Wilcoxon test (introduced in Section 2.4.2) to analyze the differences between measurements of the awake state and that of drowsy state. For a participant, if P value < 0.05, the measurement can distinguish drowsy driving [53]. When measurements calculated by UCPs were used to build drowsiness-detection models without considering individual differences, the number of participants whose drowsiness-driving can be detected by the model using unified parameters should be as many as possible. Given UCPs, we calculated measurements by UCPs and did the Wilcoxon test on individual driver's measurements. The more the participants with P value < 0.05 are, the better the UCPs are. And for a participant, the same original experimental data were used to calculate UCPs and IDBCPs, so the experimental setting had no influence on the results. Thus, through iterating various UCPs and comparing the number of drivers whose drowsiness can be detected by measurements calculated by different UCPs, we found the suitable UCPs of each measurement which can make the number of participants whose P value < 0.05 maximum.

According to reference [18, 30], we designed the value ranges of UCPs. T_1 corresponds to the duration of the drowsy state (15 s–75 s), and T_2 corresponds to the duration of the typical characteristics of drowsiness (5 s–20 s). Therefore, the value range of T_1 and T_2 is designed to be 15–75 and 5–20, respectively. The frequencies of sensors were above 20 Hz, and every second contains a lot of data, reflecting the driver's operation details. Therefore, the drowsiness characteristics can be obtained by using the related smaller T_2 to calculate measurements. And, the larger T_2 can contain the nondrowsy data and weaken the drowsiness characteristics. S_1 and S_2 affect the quantity of T_1 and T_2, respectively. Using small S_2 is convenient for selecting max drowsiness measurement value within T_1. However, if S_2 is too small, it will increase the number of T_2 and increase the amount of model computation. Also, there are gaps among typical fatigue operating features. To make sure the best quality and reduce the calculation amount, we set the range of S_1 and S_2 to be 2–5 and 7–10, respectively, to balance the amount of computation and reduce the degree of freedom. Table 2 is about the important information of nonintrusive measurements.

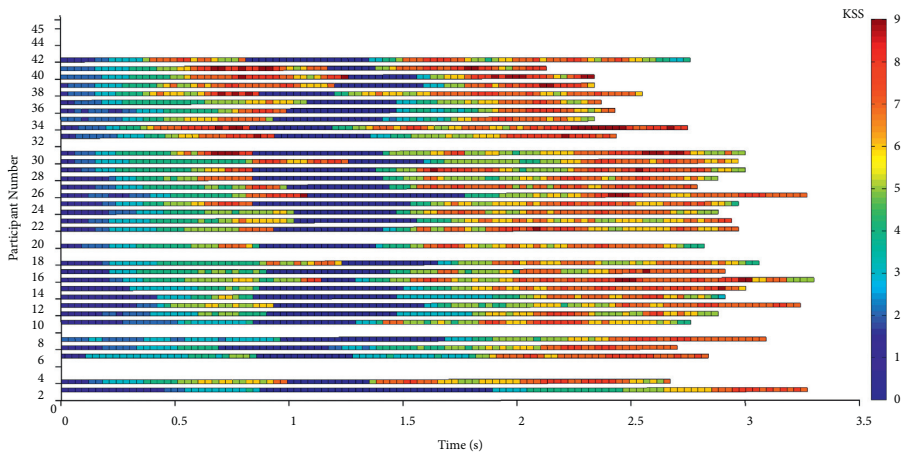


Figure 3: Graph of KSS changes of all participants over time.

Table 2: The information on nonintrusive drowsiness-related measurements.

Data source	Order	Symbol (unit)	UCPs (unit)
Mobileye	①	SDLP (m)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 4$ (s), $T_2 = 18$ (s)
Steering wheel angle sensor	②	SRR (/)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 4$ (s), $T_2 = 18$ (s), Thr = 6 (degree)
	③	SDSWM (degree/s)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 2$ (s), $T_2 = 15$ (s)
	④	VSA (/)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 4$ (s), $T_2 = 18$ (s)
Inertial navigation sensor	⑤	SDLA (m^2/s)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 4$ (s), $T_2 = 14$ (s)
	⑥	SDTV (m/s)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 4$ (s), $T_2 = 15$ (s)
Driver state sensor	⑦	MPE (/)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 3$ (s), $T_2 = 15$ (s)
	⑧	SDPE (/)	$S_1 = 6$ (s), $T_1 = 45$ (s), $S_2 = 4$ (s), $T_2 = 18$ (s)

SRR = steering reversal rate, Thr = reversal threshold, / = dimensionless measurements have no units, SDSWM = standard deviation of steering wheel movement, VSA = variation of steering wheel angle, SDLA = standard deviation of longitudinal acceleration, SDTV = standard deviation of transverse velocity, MPE = mean of PERCLOS, and SDPE = standard deviation of PERCLOS.

Calculation formulas of some complicated measurements are as follows.

② SRR measures the frequency of sampling points whose angle exceeds the wheel angle threshold, which can reflect the stability of steering wheel control [18].

$$\mathrm{SRR} = \frac{N_{\mathrm{thr}}}{N_{\mathrm{all}}}. \tag{1}$$

N_{thr} is the number of sampling points whose angle exceeds the thr and N_{all} is the total number of sampling points in the time window.

④ VSA can eliminate the influence of road curvature on steering wheel angle to some extent, which reflects the relationship between drowsiness and steering wheel angle [54].

$$\mathrm{VSA} = \frac{\mathrm{SDSWA}}{\mathrm{MSWA}}. \tag{2}$$

MSWA and SDSWA are, respectively, the mean value and standard deviation of the steering wheel angle in the time window.

⑧ SDPE describes the variation of PERCLOS in the time window, which reflects the blink frequency.

$$\mathrm{SDPE} = \sqrt{\sum_{i=1}^{n_{\mathrm{pe}}} \left(\mathrm{PE}_i - \frac{1}{n_{\mathrm{pe}}} \sum_{i=1}^{n_{\mathrm{pe}}} \mathrm{PE}_i \right)^2 \frac{1}{n_{\mathrm{pe}} - 1}}. \tag{3}$$

PE_i is the PERCLOS at sampling point i, and n_{pe} is the total number of PERCLOS sampling points in the time window. The PERCLOS is directly output by DSS. Within a certain period, DSS calculates the proportion of time the eyes are at least 80 percent closed. Then, for increasing the volume of data, DSS obtains the PERCLOS of each sampling point through interpolation. And, the sampling frequency of PERCLOS is about 60 Hz. Therefore, we can calculate the mean and standard deviation of PERCLOS in T_2.

2.4. Extraction Model of Individual Driver's Optimal Drowsiness-Detection Measurements

2.4.1. Structure of the Model. The framework for the extraction model of optimal drowsiness-detection measurements of individual drivers based on the Wilcoxon test and genetic algorithm (GA) is shown in Figure 4.

As shown in Figure 4, the model is divided into three parts. Measurements of the individual driver are input into this model to obtain IDBCPs and compose the individual driver's optimal drowsiness-detection measurement set. In part A, through performing the Wilcoxon test on measurements of the awake state and that of drowsy states, we obtain $|Z\text{-statistics}|$ ($|Z|$) representing the drowsiness-detection performance of measurements and construct the

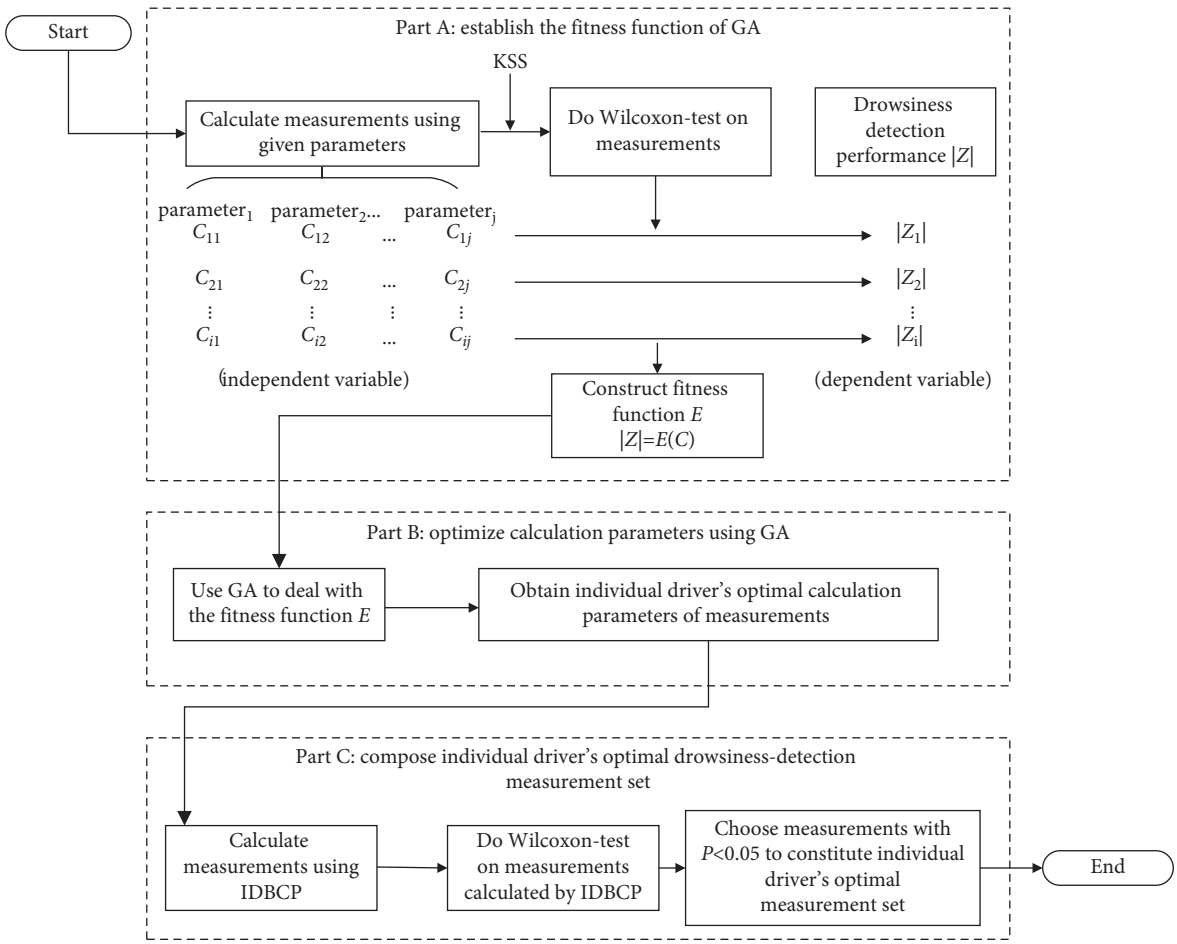


Figure 4: Extraction model of optimal drowsy driving measurements of individual drivers.

fitness function reflecting the correspondence between the calculation parameters and measurements' drowsiness-detection performance. In part B, the GA is used to optimize the fitness function and obtain the measurements' IDBCPs. In part C, we complete the Wilcoxon test on measurements that are computed by IDBCPs. We choose the measurements whose P value < 0.05 to compose the individual driver's optimal drowsiness-detection measurement set.

2.4.2. Establishing Fitness Function of GA. Firstly, the measurement samples are computed by the given calculation parameters. And, according to KSS, measurement samples are divided into samples of the awake state or that of the drowsy state.

Secondly, we perform the Wilcoxon test on measurements of the awake state and that of the drowsy state and obtain Z-statistics. Measurement sample sizes of the awake state and that of the drowsy state are different and samples are not normally distributed. Therefore, the Wilcoxon test was used to analyze the differences between the measurement samples of the awake state and that of the drowsy state. *The Wilcoxon test* is to analyze whether differences between unpaired samples from two groups are statistically significant without requesting normal distribution [53]. The bigger the $|Z|$, the greater the difference between measurement samples of the awake state and drowsy state. Thus, $|Z|$ represents the drowsiness-detection performance of measurements. The method of calculating Z-statistics is as follows.

Measurement samples in the awake driving state are A: $A_1, \ldots, A_i, \ldots, A_n$, and measurement samples in the drowsy driving state are D: $D_1, \ldots, D_j, \ldots, D_m$. The sample size of the awake state and that of the drowsy state is, respectively, n and m. Measurement samples of awake state and that of drowsy state are mixed. The mixed measurement samples are sorted in the ascending order.

$$\begin{aligned} W_A &= \sum_{i=1}^{n} R_i W_D = \sum_{j=1}^{m} R_j W_{DA} = W_A - \frac{m(m+1)}{2} W_{AD} \\ &= W_D - \frac{n(n+1)}{2}, \\ U &= \min\{W_{DA}, W_{AD}\}, \\ Z &= \frac{U - mn/2}{\sqrt{mn(n+m+1)/12}} \sim N(0,1). \end{aligned} \tag{4}$$

R_i and R_j are the rank in mixed samples of the awake sample A_i and the drowsy sample D_j, respectively.$N(0,1)$ means that the data are normally distributed.

Finally, the fitness function is established as follows:

$$|Z| = E(C). \tag{5}$$

$C(C_1, C_2, \ldots, C_j)$ is a vector consisting of independent variables and C_j was the j-th calculation parameter of the measurement. Take SRR in Table 2 as an example; the calculation parameters of SRR are S_1, T_1, S_2, T_2, and Thr. Therefore, $C(C_1, C_2, \ldots, C_j)$ of SRR is $C(S_1, T_1, S_2, T_2, \text{Thr})$. We can get the corresponding $|Z_i|$ when a set of calculation parameters $C_i(C_{i1}, C_{i2}, \ldots, C_{ij})$ is given. Therefore, the fitness function reflects the relationship between the calculation parameters and the drowsiness-detection performance ($|Z|$) of the measurement. And, the larger the fitness value ($|Z|$) is, the better the calculation parameters of the measurement are.

2.4.3. Optimizing Calculation Parameters Using GA. The genetic algorithm is a method of searching for the optimal solution based on population intelligence with advantages of simplicity, high efficiency, and good robustness [55]. Thus, we used GA to optimize the fitness function ($|Z| = E(C)$) to obtain measurements' IDBCPs. Chromosomes are individuals, and multiple individuals constitute a population. The flow chart of GA is shown in Figure 5. And the main steps of GA are introduced as follows:

Encoding measurement's calculation parameters: a chromosome consists of calculation parameters of measurements that need to be optimized. We use binary coding and decoding because of its advantages of high search efficiency and easy convergence, and convert calculation parameters into binary strings of different lengths according to the solution space and accuracy of the calculation parameters. For instance, when we take SDLP in Table 2, the calculation parameters of SDLP are S_1, T_1, S_2, T_2, and for SDLP, chromosome $= [S_1, T_1, S_2, T_2] = (6, 45, 4, 18)_{10} = (110, 101101, 100, 10010)_2$.

Calculating fitness value: the chromosome which represents measurement's calculation parameters is input into the function ($|Z| = E(C)$). Through the Wilcoxon test on measurements, the fitness value ($|Z|$) of the chromosome is obtained.

Choosing reproduction: chromosomes with larger fitness values are more likely to be selected for reproduction (crossover or mutation) [55]. The probability of the chromosome being selected equals to $|Z_i| / \sum |Z_i|$. Z_i is the fitness value of chromosome i. And, we obtain better chromosomes by roulette selection.

Crossover: two chromosomes are randomly selected to crossover according to the cross probability (P_{cr}), and then a pair of nodes are randomly selected for the exchange to generate two new chromosomes.

Mutation: a chromosome is randomly select to mutate according to the mutation probability (P_{mu}). And, then a node on its binary string is randomly selected to replace it with an allele value.

Relative process parameters of GA: learn from related research [55], parameters of GA are designed as follows, $N = 50$ (N is the population size), $P_{\text{cr}} = 0.6$, $P_{\text{mu}} = 0.01$.

Termination condition: after the genetic algebra reaches 200, GA is ended. Finally, by decoding, the binary optimal solution is converted to decimal, and IDBCPs of measurements are output.

In this paper, for fusing multiple measurements of a sample (T_1) to identify drowsiness, all measurements should have the same S_1 and T_1. First, some groups (S_1, T_1) are generated by enumeration. Then, all measurements use the same (S_1, T_1), and the GA is used to optimize the remaining calculation parameters to obtain the maximum $|Z_{\max}|$ of each measurement under the same (S_1, T_1). Finally, (S_1, T_1) with the largest sum of all measurements' $|Z_{\max}|$ is chosen as the best (S_1, T_1) of all measurements. Finally, (S_1, T_1) with the largest sum of all measurements' $|Z_{\max}|$ is chosen as the best (S_1, T_1) of all measurements, which together with other optimal parameters constitutes the measurements' IDBCPs.

2.4.4. Composing Individual Driver's Optimal Drowsiness-Detection Measurement Set. Although we optimize the calculation parameters of individual driver's measurements in part B, whether the measurements calculated by IDBCPs can effectively detect drowsy driving needs further test. Thus, we perform Wilcoxon test on measurements calculated by IDBCPs, and then measurements with P-value < 0.05 are selected to compose the optimal drowsiness-detection measurement set of the individual driver.

2.5. Validation Model of Individual Driver's Measurements Calculated by IDBCPs. For verifying the drowsiness-detection advantages of measurements' IDBCPs, for the individual driver, the multiple nonintrusive measurements of individual drivers were integrated to build driver-specific drowsiness-detection models based on the Fisher discriminant algorithm. We, respectively, used drowsiness-detection measurements calculated by UCPs and IDBCPs to build two kinds of driver-specific drowsiness-detection models named DF_U and DF_I. The drowsiness-detection advantages of using IDBCPs are illustrated by comparing drowsiness-detection accuracies of DF_U and DF_I.

Fisher discriminant algorithm is an effective classification method, which simplifies the problem by projecting the points in the high-dimensional space [56].

$$\begin{gathered} \begin{bmatrix} \text{DS}_1 \\ \text{DS}_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} C^T + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \\ P_i = \frac{e^{\text{DS}_i}}{e^{\text{DS}_1} + e^{\text{DS}_2}}, \quad i = 1, 2. \end{gathered} \tag{6}$$

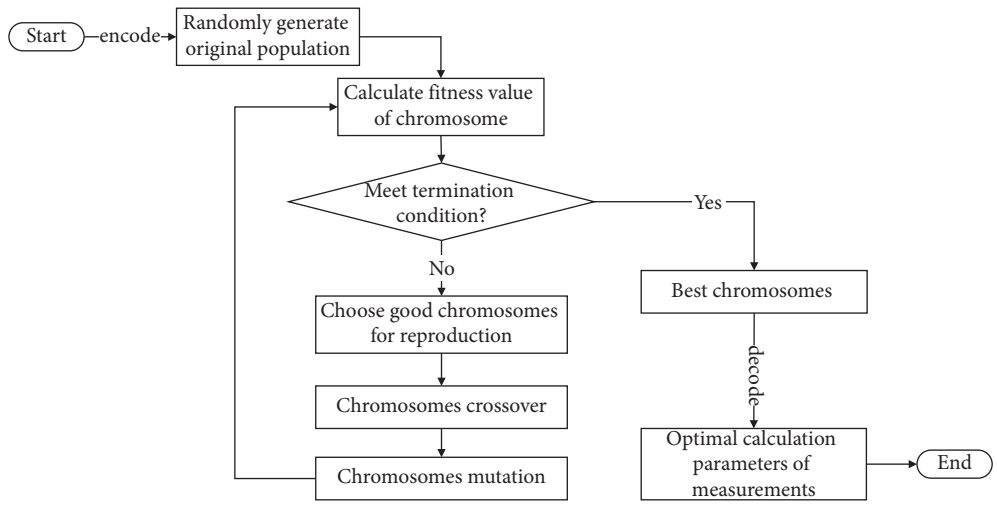


Figure 5: The flow chart of GA.

DS_1 and DS_2 are the function values of the awake state and drowsy state, respectively. $\alpha(\alpha_1, \alpha_2, \ldots, \alpha_6), \beta(\beta_1, \beta_2, \ldots, \beta_6), b_1, b_2$ are the algorithm coefficient, which is obtained by using training samples to train the model. $P_i\,(i = 1, 2)$ is the drowsiness-detection measurement of individual drivers such as SDLP and SRR. $P_i\,(i = 1, 2)$ is the probability that the sample belongs to the awake state or drowsy state. The driving state with a higher probability is the identification result of the model.

3. Results

To save paper space, 6 participants (No. 5, No. 11, No. 14, No. 18, No. 23, and No. 29) were randomly selected as examples to display the related results in this section.

3.1. Driver Fingerprinting Differences of Measurements Using UCPs. We took SDLP as an example to display driver fingerprinting differences of measurements calculated by UCPs. Figure 6 shows the radar charts composed of the quartiles of the SDLP of 6 participants in the awake state and drowsy state.

In Figure 6, all closed lines were not circular but irregular polygons. It meant that the distributions of SDLP calculated by the same calculation parameters of 6 participants were different, which was the outcome of driver fingerprinting differences in lane control behavior. There were differences in the median of SDLP among participants in the awake state and drowsy state. For instance, in the awake state, the medians of No. 11 and No. 18 were maximum (0.35) and minimum (0.20), respectively. And, in the drowsy state, the medians of No. 11 and No. 18 were maximum (0.41) and minimum (0.24), respectively. Besides, the ranges of SDLP among participants were also different. In the awake state, the differences between the upper and lower quartiles of No. 11 and No. 18 were the maximum (0.18) and minimum (0.07), respectively. In the drowsy state, the differences between the upper and lower quartiles of No. 14 and No. 18 were the maximum (0.21) and minimum (0.12), respectively.

3.2. Optimal Drowsiness-Detection Measurements of Individual Participants. All samples were divided into the training sample set and testing sample set, with the proportions of 70% and 30%, respectively. Using stratified random sampling, 70% of the data in awake and drowsy states were, respectively, selected to form the training sample set, and the remaining data were the testing sample set. The stratified random sampling was repeated three times, and the final result was taken as the average value of repeated calculations. Table 3 presents the results of models in Figure 4 of the example participants. And in Table 3, $|Z|$ is $|Z$-statistics$|$ of measurements calculated by IDBCPs, and * means P value < 0.05.

As shown in Table 3, most measurements using IDBCPs could distinguish drowsy driving (P value < 0.05). However, for some participants, although the measurements' calculation parameters were optimized, the measurements calculated by IDBCPs could not still distinguish drowsy driving (P value < 0.05). For example, the P value of SDSWM of No. 11 exceeded 0.05. Therefore, SDSWM could not be selected to compose an optimal drowsiness-detection measurement set of No. 11.

Participant No. 14 was chosen as an example to display the optimization process of SRR's calculation parameters. The training samples of SRR were input into the model in Figure 4. As is shown in Figure 7, the fitness value gradually increased with iterations, which indicated that the model could optimize the calculation parameters and improve the drowsiness-identification performance of measurements.

3.3. Comparison of Measurements Calculated by IDBCPs and UCPs. The testing samples were used to analyze the change of drowsiness-detection performance of measurements calculated by IDBCPs. In Table 4, we took SDSWM as an example, and $|Z|$ of SDSWM calculated by UCPs ($|Z|$-UCPs) and IDBCPs ($|Z|$-IDBCPs) were, respectively, listed.

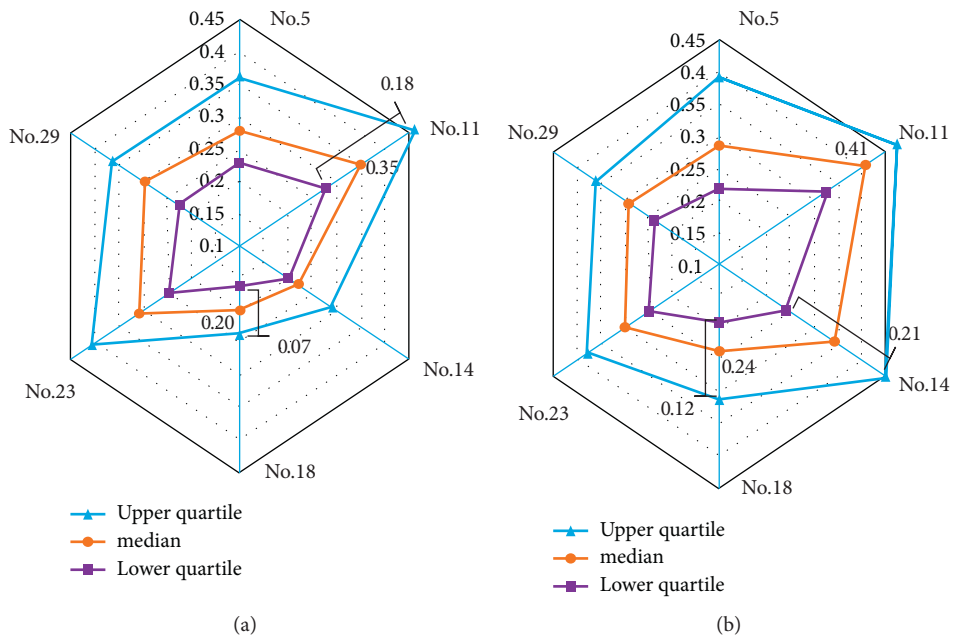


Figure 6: The SDLP calculated by the UCPs radar chart of 6 participants. (a) SDLP of the awake driving state. (b) SDLP of the drowsy driving state.

Table 3: The IDBCPs and $|Z|$ of measurements for example participants.

Participant number		No. 5 (17)	No. 11 (23)	No. 14 (31)	No. 18 (30)	No. 23 (35)	No. 29 (41)
Measurements		$S_1 = 7,\ T_1 = 62$	$S_1 = 7,\ T_1 = 66$	$S_1 = 7,\ T_1 = 54$	$S_1 = 7,\ T_1 = 56$	$S_1 = 8,\ T_1 = 65$	$S_1 = 7,\ T_1 = 50$
SDLP	S_2	3	4	5	2	3	5
	T_2	20	15	20	18	14	17
	$\|Z\|$	4.65*	4.44*	6.84*	6.54*	2.83*	1.17
SRR	S_2	5	2	3	3	2	2
	T_2	20	20	20	18	15	16
	Thr	5	6	6	7	6	7
	$\|Z\|$	4.34*	2.34*	3.23*	2.64*	4.07*	6.03*
SDSWM	S_2	3	5	5	2	2	2
	T_2	20	14	19	19	14	20
	$\|Z\|$	7.05*	1.64	7.51*	6.40*	9.30*	7.65*
VSA	S_2	3	5	3	5	5	5
	T_2	20	18	14	17	20	14
	$\|Z\|$	3.94*	2.95*	9.46*	2.05*	2.05*	3.93*
SDLA	S_2	3	3	3	4	3	2
	T_2	19	20	14	14	18	20
	$\|Z\|$	5.89*	5.42*	2.57*	3.91*	2.59*	3.51*
SDTV	S_2	3	5	3	2	2	2
	T_2	20	17	16	14	15	20
	$\|Z\|$	8.67*	1.93	7.54*	10.84*	4.75*	3.82*
MPE	S_2	3	2	2	2	4	4
	T_2	14	16	16	14	14	19
	$\|Z\|$	6.22*	4.32*	10.79*	5.14*	16.63*	2.83*
SDPE	S_2	3	5	4	3	3	4
	T_2	20	14	20	20	18	14
	$\|Z\|$	3.92*	3.02*	5.54*	1.87	8.93*	1.73

In Table 4, for every participant, $|Z|$-IDBCPs was bigger than $|Z|$-UCPs, which meant that using IDBCPs to calculate SDSWM could improve the contribution of SDSWM for drowsiness detecting. However, the improvement extent for participants was various, that of No. 18 and No. 29 was, respectively, maximum (2.33) and minimum (0.66). It could

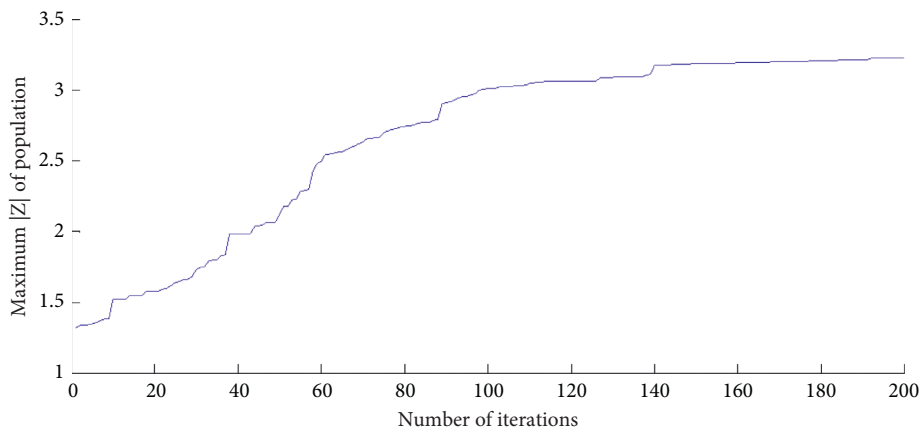


Figure 7: The maximum $|Z|$ of SRR in iterations of GA.

Table 4: $|Z|$ of SDSWM calculated by UCPs and IDBCPs, respectively.

Participant $\|Z\|$	No. 5	No. 11	No. 14	No. 18	No. 23	No. 29
$\|Z\|$_IDBCPs	4.60	1.41	7.37	6.16	8.16	6.49
$\|Z\|$_UCPs	3.71	0.67	6.59	3.83	6.95	5.84
$\triangle\|Z\|$	0.89	0.74	0.78	2.33	1.21	0.66

$\triangle|Z| = |Z|$_IDBCPs-$|Z|$_UCPs.

be seen that the necessaries of using IDBCPs for different participants were different, and it was more urgent for No. 18 to use IDBCPs to calculate SDSWM.

To study the distribution differences between measurements calculated by UCPs and that calculated by IDBCPs, in Figure 8, we chose No. 18 to, respectively, draw the boxplots of SDSWM calculated by UCPs and IDBCPs in the awake state and drowsy state.

In Figure 8, whether the SDSWM was calculated by UCPs or IDBCPs, the median of the SDSWM in the drowsy state was greater than that in the awake state. Although there is still an overlap between SDSWM of awake state and that of drowsy state, by performing the Wilcoxon test on SDSWM of awake state and that of drowsy state, P value was less than 0.05, which illustrated differences between SDSWM in waking state and SDSWM in the drowsy state was statistically significant, and drowsy driving could be distinguished by using SDSWM. It indicated that the drowsiness impaired the ability to control the steering wheel and reduced the stability of the steering wheel movement, which was consistent with previous study conclusions [12]. Besides comparing with SDSWM calculated by UCPs, the overlap decreases when using IDBCPs to calculate SDSWM. The difference between the median of SDSWM calculated by UCPs of the awake state and that of the drowsy state was 0.05, while the difference between the median of SDSWM calculated by IDBCPs of the awake state and that of the drowsy state was 0.07. The result showed that when measurements were calculated by IDBCPs, distribution differences between measurements in the awake state and that in the drowsy state were more significant, and its drowsiness-detection performance was stronger.

3.4. Verification of Individual Driver's Optimal Drowsiness-Detection Measurements. Drowsiness-detection advantages of measurements calculated by IDBCPs were verified by comparing drowsiness-detection accuracies of DF_U and DF_I. For the individual participant, the optimal drowsiness-detection measurements calculated by IDBCPs were fused to build DF_I, and the same drowsiness-detection measurements calculated by UCPs were also fused to build DF_U. For individual drivers, according to stratified sampling, 80% of the samples were, respectively, extracted from measurements in the awake state and that in the drowsy state to compose the training sample set, and the rest constituted the testing sample set. The training samples were used to train driver-specific drowsiness-detection models based on the Fisher discriminant algorithm. In Table 5, for DF_U and DF_I, we displayed the mean, standard deviation, minimum, and maximum of drowsiness-detection results across all participants.

As shown in Table 5, the related drowsiness-detection results of DF_U were better than that of DF_I. The mean drowsiness-detection accuracy across 35 participants of DF_I was 91.06%, which was higher than that of DF_U (85.25%). The results indicated building driver-specific models based on measurements calculated by IDBCPs could improve drowsiness-detection accuracies.

For every participant, Figure 9 shows the comparison of the drowsiness-detection accuracy of DF_U and DF_I. In Figure 9, for each example participant, the accuracy of DF_I was higher than that of DF_U. Besides, for different participants, the accuracy improvements equalling the accuracy of DF_I minus the accuracy of DF_U were various. The accuracy improvement of No. 6 was maximum with the value of 8.11% (DF_U = 81.73%, DF_I = 89.84%), whereas that of No. 34 was minimum with the value of 1.56% (DF_U = 86.89%, DF_I = 88.45%). The possible interpretation was that the IDBCPs of some participant were similar to UCPs and the improvement of the drowsiness-detection performance of measurements using IDBCPs is limited.

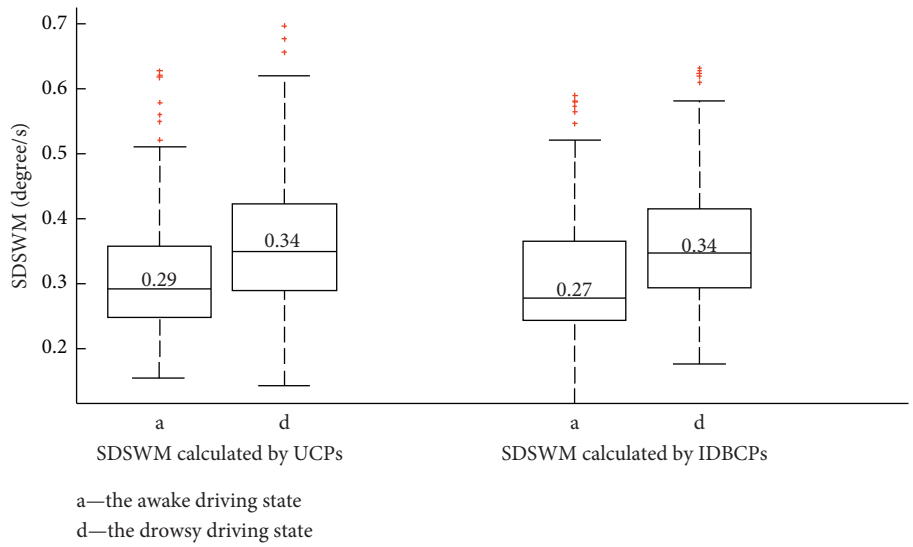


Figure 8: The boxplots of SDSWM calculated by UCPs and IDBCPs.

Table 5: Results of drowsiness-detection across all participants.

		Mean (%)	Standard deviation (%)	Minimum (%)	Maximum (%)
DF_U	Accuracy	85.25	2.35	79.4	90.2
	Sensitivity	87.5	2.43	81.33	91.46
	Specificity	84.15	2.54	78.32	89.6
	*F*1	79.56	3.1	71.76	85.71
DF_I	Accuracy	91.06	2.93	81.63	95.97
	Sensitivity	93.39	3.06	83.95	97.78
	Specificity	89.92	3.02	80.49	95.6
	*F*1	87.38	3.91	75.14	94.51

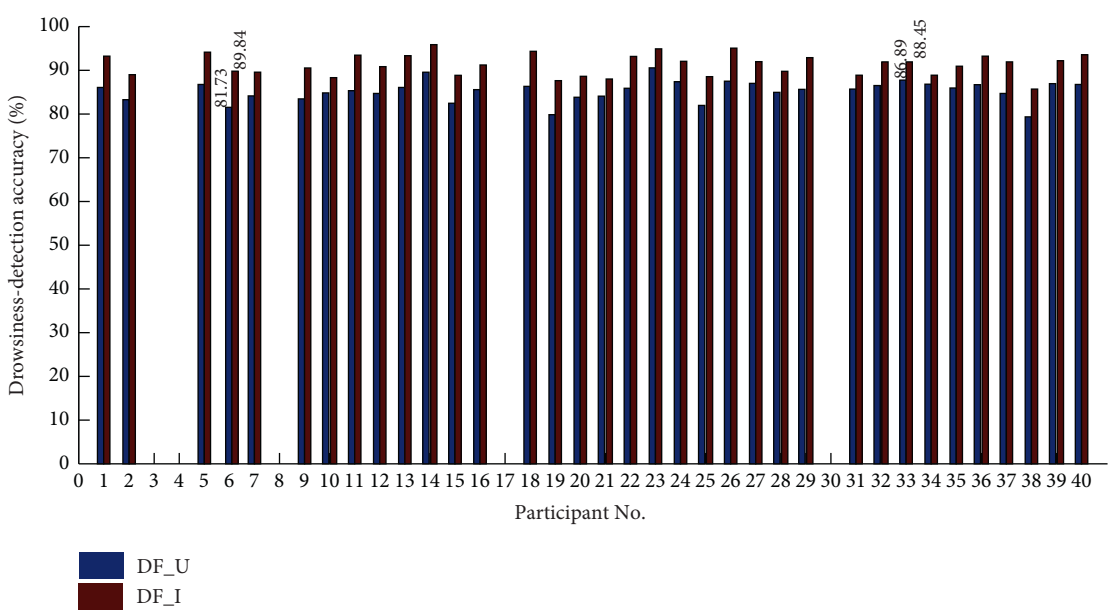


Figure 9: The drowsiness-detection accuracy of DF_U and DF_I.

4. Discussion

By summarizing the above results, important insights and contributions to the development of drowsiness-detection methods considering driver fingerprinting differences were obtained. There are individual differences in data distribution of measurements using UCPs (Figure 6), which is consistent with the previous research [37]. The reason for this phenomenon is that there are differences in drivers' raw driving behavior data, which is the typical manifestation of driver fingerprinting. Similar to the studies [33, 34], it is indicated that individual drivers have unique characteristics in driving behavior and physiology, and there are differences in the driver fingerprints among the drivers. Drowsiness-detection thresholds are the average of all drivers when all drivers' mixed measurement data are used as a whole to train models. And, driver fingerprinting differences make drowsiness-detection of the individual driver to suffer from systematic error. For example, the individual driver's drowsiness-detection threshold is below the average threshold (like No. 18 in Figure 6), the driver's drowsiness cannot be detected when measurements exceed this individual driver's drowsiness-detection threshold but not reach the average threshold of all drivers. Therefore, it is worthy to study accurate drowsiness-detection methods considering driver fingerprinting differences.

The major achievement of this paper is to propose a model to optimize individual driver's calculation parameters of measurements considering driver fingerprinting differences and extract individual driver's optimal drowsiness-detection measurements. It is found that the IDBCPs of measurements among drivers are significantly different (Table 3), which is attributed to driver fingerprinting differences among drivers. Figure 8 confirms that using IDBCPs to calculate individual drivers' measurements can enhance the contribution of measurements for detecting drowsy driving. Similar to existing studies [18, 32, 36], every driver has unique driver fingerprinting characteristics, and the effect of drowsiness on the driving behavior of each driver is also different [10], which causes many individual differences in the raw data related to the drowsiness. For example, the time length of the drowsy state, duration of typical drowsy driving behavior characteristics, and distribution of drowsiness-related driving behavior data of each driver are different. Without considering the driver fingerprinting differences, using UCPs to calculate measurements causes measurements to fail to accurately reflect drowsy driving behavior characteristics of individual drivers, which limits the drowsiness-detection performance of nonintrusive measurements at the individual driver level. Thus, it is suggested to optimize measurements' calculation parameters of individual drivers to obtain IDBCPs suitable for individual driver's driver fingerprinting characteristics at different drowsiness levels, which can strengthen the sensitivity of measurements to drowsiness.

Comparison results of drowsiness-detection accuracy in Figure 9 illustrate that DF_I outperforms DF_U, which verifies the drowsiness-detection advantages of using IDBCPs to calculate individual driver's measurements. The reason is that DF_I utilizes measurements calculated by IDBCPs with stronger drowsiness-identification performance to build models. The previous study establishes the driver-specific drowsiness-detection model in a simulated driving environment using measurements calculated by UCPs [39]. Although the accuracy of DF_U (84.16%) is lower than that of previous research (88.60%), there are many interferences in the real driving environment increasing complexities and difficulties of detecting drowsiness. Thus, the drowsiness-detection accuracy of DF_U is ideal. Moreover, the average drowsiness-detection accuracy of DF_I is higher than that of the previous research [39], and accuracies of some drivers reach more than 93.00% (No. 14 and No. 23 in Figure 9). It is demonstrated that the drowsiness of individual drivers can be reliably detected with higher accuracy using nonintrusive measurements calculated by IDBCPs, which is also inferred in the previous study [18].

As mentioned in the existing studies, the calculation and selection of measurements is an important factor affecting drowsiness-detection accuracy [26, 39]. Therefore, the model and results of this paper can be applied to develop personalized anti-drowsiness systems in real-world conditions. Individual driver's optimal drowsiness-detection measurements calculated by IDBCPs should be used to train driver-specific drowsiness-detection models. According to the results in Figure 9, the drowsiness-detection accuracy can be improved by using measurements calculated by IDBCPs having higher drowsiness-detection performance. It is worth highlighting that for some drivers whose accuracy improvement of DF_I is high, it is more indispensable to build the driver-specific drowsiness-detection models using measurements calculated by IDBCPs.

5. Conclusions

Due to driver fingerprinting differences, the IDBCPs matching individual driver's behavior characteristics may be very different from UCPs, and using IDBCPs to calculate measurements enhances measurements' drowsiness-detection performance for individual drivers. Therefore, the purpose of this paper was to propose a model to optimize measurements' calculation parameters of individual drivers. Firstly, the naturalistic driving data and KSS of 35 participants were collected through field experiments. Eight nonintrusive measurements related to drowsiness were chosen and computed using the double-layer slide time window. Then, based on Wilcoxon test and GA, we established a model (Figure 4) to extract individual drivers' optimal drowsiness-detection measurements. Finally, based on the Fisher discriminant algorithm, the two kinds of driver-specific drowsiness-detection models were built using measurements calculated by UCPs and IDBCPs, respectively. And, drowsiness-detection advantages of using IDBCPs to calculate measurements were illustrated.

In the present paper, we obtained individual driver's optimal drowsiness-detection measurements calculated by IDBCPs, and the results of example participants are shown

in Table 3. Figure 6 verifies that there are individual differences in nonintrusive measurements calculated by UCPs, which is attributed to the driver fingerprinting differences among drivers. This paper verifies that for individual drivers, the drowsiness-detection performance of measurements calculated by IDBCPs is better than that of measurements calculated by UCPs. In addition, the results showed that the average drowsiness-detection accuracy of DF_I was 6.25% higher than that of DF_U, which indicated that using nonintrusive measurements calculated by IDBCPs to establish the driver-specific drowsiness-detection models improved drowsiness-detection accuracy at the individual driver level.

Admittedly, there are some limitations in this paper. For instance, we used participants' self-report KSS as ground truth for drowsiness, and the drowsy levels were relatively subjective. Besides, the amount of experimental data of some individual drivers was not very sufficient which led to insufficient model training and not very high drowsiness-detection accuracy for some participants. For a new user, firstly, we need to label the data with drowsiness and awakeness, which is cumbersome and a limitation in practice. However, for some special drivers such as dangerous goods transport drivers and long-distance bus drivers, it is still very worthwhile to establish a personalized model based on measurements calculated by IDBCPs to improve the drowsiness-detection accuracy.

In the future, we will adopt the more objective and precise standard as the ground truth for drowsiness such as electroencephalogram. And, higher performance models such as artificial neural networks and Bayesian networks will be used to build drowsiness-detection models to reach better accuracies. Ultimately, this research can provide references for the calculation of drowsiness measurements using naturalistic driving data, and guide the establishment of drowsiness-detection models considering driver fingerprinting differences, which can accelerate the development of personalized anti-drowsiness driving active safety systems in vehicles. Meanwhile, the efforts highlight the advantages of studying driver fingerprinting differences and promote the application of driver fingerprinting in the field of dangerous driving behavior.

Data Availability

The raw data used in this study are available from the corresponding author on request.

Conflicts of Interest

The data indicated in the findings have not been made available due to data privacy.

Acknowledgments

This research was supported by the National Key Research Program of China (2019YFB1600800); National Natural Science Foundation of China (52072289 and U1764262).

References

[1] G. Adamos and E. Nathanail, "Testing the effectiveness of objective and subjective predictors of driving behavior under fatigue," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 8, pp. 343–352, 2019.

[2] G. Zhang, K. K. W. Yau, X. Zhang, and Y. Li, "Traffic accidents involving fatigue driving and their extent of casualties," *Accident Analysis & Prevention*, vol. 87, pp. 34–42, 2016.

[3] Akerstedt, "Consensus statement: fatigue and accidents in transport operations," *Journal of Sleep Research*, vol. 9, no. 4, p. 395, 2000.

[4] A. Moradi, S. Nazari, and K. Rahmani, "Sleepiness and the risk of road traffic accidents: a systematic review and meta-analysis of previous studies," *Transportation Research Part F: Traffic Psychology Behaviour*, vol. 65, pp. 620–629, 2018.

[5] A. Williamson, D. A. Lombardi, S. Folkard, J. Stutts, T. K. Courtney, and J. L. Connor, "The link between fatigue and safety," *Accident Analysis & Prevention*, vol. 43, no. 2, pp. 498–515, 2011.

[6] NHTSA, *A Sleep at the Wheel: A National Compendium of Efforts to Eliminate Drowsy Driving*, U.S. Department of Transportation, Washington, DC, USA, 2017.

[7] Fatigue, 2017, https://ec.europa.eu/transport/roadsafety/.

[8] M. L. Lee, M. E. Howard, W. J. Horrey et al., "High risk of near-crash driving events following night-shift work," *Proceedings of the National Academy of Sciences*, vol. 113, no. 1, pp. 176–181, 2015.

[9] D. Dawson, A. C. Reynolds, H. P. A. Van Dongen, and M. J. W. Thomas, "Determining the likelihood that fatigue was present in a road accident: a theoretical review and suggested accident taxonomy," *Sleep Medicine Reviews*, vol. 42, pp. 202–210, 2018.

[10] A. Anund, G. Kecklund, B. Peters, and T. Åkerstedt, "Driver sleepiness and individual differences in preferences for countermeasures," *Journal of Sleep Research*, vol. 17, no. 1, pp. 16–22, 2008.

[11] C. C. Liu, S. G. Hosking, and M. G. Lenné, "Predicting driver drowsiness using vehicle measures: recent insights and future challenges," *Journal of Safety Research*, vol. 40, no. 4, pp. 239–245, 2009.

[12] G. Sikander and S. Anwar, "Driver fatigue detection systems: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2018.

[13] Q. He, W. Li, X. Fan, and Z. Fei, "Driver fatigue evaluation model with integration of multi-indicators based on dynamic Bayesian network," *IET Intelligent Transport Systems*, vol. 9, no. 5, pp. 547–554, 2014.

[14] M. Doudou, A. Bouabdallah, and V. Berge-Cherfaoui, "Driver drowsiness measurement technologies: current research, market solutions, and challenges," *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 2, pp. 297–319, 2019.

[15] J. Vicente, P. Laguna, A. Bartra, and R. Bailón, "Drowsiness detection using heart rate variability," *Medical, & Biological Engineering & Computing*, vol. 54, no. 6, pp. 927–937, 2016.

[16] F. You, Y. Gong, H. Tu, J. Liang, and H. Wang, "A fatigue driving detection algorithm based on facial motion information entropy," *Journal of Advanced Transportation*, vol. 2020, Article ID 8851485, 17 pages, 2020.

[17] Z. Li, S. Li, R. Li, B. Cheng, and J. Shi, "Online detection of driver fatigue using steering wheel angles for real driving conditions," *Sensors*, vol. 17, no. 3, p. 495, 2017.

[18] H. Zhang, C. Wu, Z. Huang, X. Yan, and T. Z. Qiu, "Sensitivity of lane position and steering angle measurements to driver fatigue," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2585, no. 1, pp. 67–76, 2016.

[19] A. S. Zandi, A. Quddus, L. Prest, and F. J. E. Comeau, "Non-intrusive detection of drowsy driving based on eye tracking data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 6, pp. 247–257, 2019.

[20] G. Sikander and S. Anwar, "Driver fatigue detection systems: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2019.

[21] A. Němcová, V. Svozilová, and K. Bucsuházy, "Multimodal features for detection of driver stress and fatigue: review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3214–3233, 2020.

[22] S. Soares, S. Ferreira, and A. Couto, "Driving simulator experiments to study drowsiness: a systematic review," *Traffic Injury Prevention*, vol. 1, pp. 1–9, 2020.

[23] Z. Li, S. Li, R. Li, B. Cheng, and J. Shi, "Driver fatigue detection using approximate entropic of steering wheel angle from real driving data," *International Journal of Robotics and Automation*, vol. 32, no. 3, pp. 291–298, 2017.

[24] T. Wakita, "Driver identification using driving behavior signals," *IEICE-Transactions on Info and Systems*, vol. E89-D, no. 3, pp. 1188–1194, 2006.

[25] R. Rossi, M. Gastaldi, and G. Gecchele, "Analysis of driver task-related fatigue using driving simulator experiments," *Procedia-Social and Behavioral Sciences*, vol. 20, pp. 666–675, 2011.

[26] K. Henni, N. Mezghani, C. Gouin-Vallerand, P. Ruer, Y. Ouakrim, and É. Vallières, "Feature selection for driving fatigue characterization and detection using visual- and signal-based sensors," *Applied Informatics*, vol. 5, no. 1, pp. 1–15, 2018.

[27] R. Feng, G. Zhang, and B. Cheng, "An on-board system for detecting driver drowsiness based on multi-sensor data fusion using Dempster-Shafer theory," in *Proceedings of the International Conference on Networking*, Valencia, Spain, 2009.

[28] A. D. McDonald, C. Schwarz, J. D. Lee, and T. L. Brown, "Real-time detection of drowsiness related lane departures using steering wheel angle," *Proceedings of the Human Factors and Ergonomics Society-Annual Meeting*, vol. 56, no. 1, pp. 2201–2205, 2012.

[29] Q. Cheng, W. Wang, X. Jiang, S. Hou, and Y. Qin, "Assessment of driver mental fatigue using facial landmarks," *IEEE Access*, vol. 7, pp. 150423–150434, 2019.

[30] X. Zhang, B. Cheng, and R. Feng, "Real-time detection of driver drowsiness based on steering performance," *Journal of Tsinghua University*, vol. 7, pp. 1072–1076, 2010.

[31] E. C. Chua, S. C. Yeo, I. T. Lee et al., "Individual differences in physiologic measures are stable across repeated exposures to total sleep deprivation," *Physiological reports*, vol. 2, no. 9, pp. 1–16, 2014.

[32] A. Bouhoute, R. Oucheikh, K. Boubouh, and I. Berrada, "Advanced driving behavior analytics for an improved safety assessment and driver fingerprinting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2171–2184, 2019.

[33] S. Ezzini, I. Berrada, and M. Ghogho, "Who is behind the wheel? Driver identification and fingerprinting," *Journal of Big Data*, vol. 5, no. 1, 2018.

[34] E. Miro, T. Alex, K. Karl, and T. Kohno, "Automobile driver fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 1, pp. 34–50, 2016.

[35] I. d. Campo, R. Finker, M. V. Martínez, J. Echanobe, and F. Doctor, "A real-time driver identification system based on artificial neural networks and cepstral analysis," in *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 1848–1855, 2014.

[36] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, "Automobile driver fingerprinting: a new machine learning based authentication scheme," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1417–1426, 2020.

[37] M. Ingre, T. Akerstedt, B. Peters, A. Anund, and G. Kecklund, "Subjective sleepiness, simulated driving performance and blink duration: examining individual differences," *Journal of Sleep Research*, vol. 15, no. 1, pp. 47–53, 2006.

[38] P. Thiffault and J. Bergeron, "Fatigue and individual differences in monotonous simulated driving," *Personality and Individual Differences*, vol. 34, no. 1, pp. 159–176, 2003.

[39] X. Wang and C. Xu, "Driver drowsiness detection based on non-intrusive metrics considering individual specifics," *Accident Analysis & Prevention*, vol. 95, no. part B, pp. 350–357, 2016.

[40] W. B. Verwey and D. M. Zaidel, "Predicting drowsiness accidents from personal attributes, eye blinks and ongoing driving behaviour," *Personality and Individual Differences*, vol. 28, no. 1, pp. 123–142, 2000.

[41] R. Yan, C. Wu, and Y. Wang, "Exploration and evaluation of individual difference to driving fatigue for high-speed railway: a parametric SVM model based on multidimensional visual cue," *IET Intelligent Transport Systems*, vol. 12, no. 6, pp. 504–512, 2018.

[42] M. Ingre, T. Akerstedt, B. Peters, A. Anund, G. Kecklund, and A. Pickles, "Subjective sleepiness and accident risk avoiding the ecological fallacy," *Journal of Sleep Research*, vol. 15, no. 2, pp. 142–148, 2010.

[43] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1139–1152, 2014.

[44] C. S. Silveira, J. S. Cardoso, A. L. Lourenço, and C. Ahlström, "Importance of subject-dependent classification and imbalanced distributions in driver sleepiness detection in realistic conditions," *IET Intelligent Transport Systems*, vol. 13, no. 2, pp. 347–355, 2018.

[45] A. Persson, H. Jonasson, I. Fredriksson, U. Wiklund, and C. Ahlstrom, "Heart rate variability for classification of alert versus sleep deprived drivers in real road driving conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 99, pp. 1–10, 2020.

[46] F. You, X. Li, Y. Gong, H. Wang, and H. Li, "A real-time driving drowsiness detection algorithm with individual differences consideration," *IEEE Access*, vol. 7, pp. 179396–179408, 2019.

[47] W. Chu, C. Wu, and H. Zhang, "Driver behavior model and its application in driver fatigue identification," *China Safety Science Journal*, vol. 28, no. 6, pp. 43–48, 2018.

[48] C. J. d. Naurois, C. Bourdin, C. Bougard, and J.-L. Vercher, "Adapting artificial neural networks to a specific driver enhances detection and prediction of drowsiness," *Accident Analysis Prevention*, vol. 121, pp. 118–128, 2018.

[49] Y. Liang, W. J. Horrey, M. E. Howard et al., "Prediction of drowsiness events in night shift workers during morning

driving," *Accident Analysis & Prevention*, vol. 126, pp. 105–114, 2019.

[50] A. Å. Miley, G. Kecklund, and T. Åkerstedt, "Comparing two versions of the Karolinska sleepiness scale (KSS)," *Sleep and Biological Rhythms*, vol. 14, no. 3, pp. 257–260, 2016.

[51] H. Singh and A. Kathuria, "Analyzing driver behavior under naturalistic driving conditions: a review," *Accident Analysis & Prevention*, vol. 150, Article ID 105908, 2021.

[52] T. a. Åkerstedt, B. A. C. Anund, J. b. Axelsson, and G. Kecklund, "Subjective sleepiness is a sensitive indicator of insufficient sleep and impaired waking function," *Journal of Sleep Research*, vol. 3, pp. 240–252, 2014.

[53] T. Moriyama and Y. Maesono, "Smoothed alternatives of the two-sample median and Wilcoxon's rank sum tests," *Statistics*, vol. 52, no. 5, pp. 1096–1115, 2018.

[54] L. Jin, K. Li, Q. Niu, and L. L. Gao, "A new method for detecting driver fatigue using steering performance," *Journal of Transport Information and Safety*, vol. 32, no. 5, pp. 103–107, 2014.

[55] K. Dejong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, University of Michigan, Arbor, MI, USA, 1975.

[56] B. Tavakkol, M. K. Jeong, and S. L. Albin, "Measures of scatter and fisher discriminant analysis for uncertain data," *IEEE Transactions on Systems, Man, Cybernetics: Systems*, vol. 51, no. 3, pp. 1690–1703, 2019.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4292778, 10 pages
https://doi.org/10.1155/2021/4292778

Research Article

Flight Delay Classification Prediction Based on Stacking Algorithm

Jia Yi,[1] Honghai Zhang,[1] Hao Liu,[2] Gang Zhong,[1] and Guiyi Li[1]

[1]*College of Civil Aviation, Nanjing University of Aeronautics&Astronautics, Nanjing 211106, China*
[2]*College of Science, Nanjing University of Aeronautics&Astronautics, Nanjing 211106, China*

Correspondence should be addressed to Honghai Zhang; zhh0913@163.com

Received 2 June 2021; Revised 19 July 2021; Accepted 11 August 2021; Published 18 August 2021

Academic Editor: Chi-Hua Chen

With the development of civil aviation, the number of flights keeps increasing and the flight delay has become a serious issue and even tends to normality. This paper aims to prove that Stacking algorithm has advantages in airport flight delay prediction, especially for the algorithm selection problem of machine learning technology. In this research, the principle of the Stacking classification algorithm is introduced, the SMOTE algorithm is selected to process imbalanced datasets, and the Boruta algorithm is utilized for feature selection. There are five supervised machine learning algorithms in the first-level learner of Stacking including KNN, Random Forest, Logistic Regression, Decision Tree, and Gaussian Naive Bayes. The second-level learner is Logistic Regression. To verify the effectiveness of the proposed method, comparative experiments are carried out based on Boston Logan International Airport flight datasets from January to December 2019. Multiple indexes are used to comprehensively evaluate the prediction results, such as Accuracy, Precision, Recall, $F1$ Score, ROC curve, and AUC Score. The results show that the Stacking algorithm not only could improve the prediction accuracy but also maintains great stability.

1. Introduction

Airports are significant nodes of air transportation. The number of airport flight delays has been on increase in recent years. Delayed flights are defined by the Federal Aviation Administration when they arrive or depart more than 15 minutes later than scheduled. In 2019, the arrival delay rate is 19.2% and the departure delay rate is 18.18% in the United States [1]. Flight delays can cause many negative effects, such as passengers' inconvenience, increased airport pressure, and airline losses [2]. Effective flight delay prediction could provide support for flight plan and emergency plan formulation, reduce the economic loss, and alleviate the negative impact. The Bureau of Transportation Statistics has recorded the nationwide flight operation data in the United States which provides valuable and reliable datasets for study flight delay issues. Meanwhile, with the development of artificial intelligence, machine learning technology has been widely used in airport flight delay prediction. Machine learning technology involves multiple disciplines, such as probability, statistics, and computer science [3]. Machine learning can break the limitations of mathematical formulas and improve the accuracy of flight delay prediction. In general, machine learning technology can be roughly divided into supervised learning, unsupervised learning, deep learning, reinforcement learning, and ensemble learning. Each of these learning methods has its characteristics. We should select the appropriate methods and algorithms to carry on research. Poorly performing algorithms not only cannot gain accurate results but also wastes computing power. Therefore, algorithm selection is an important process in machine learning technology. This paper aims to provide an applicable flight delay classification prediction method, especially for solving algorithm selection problems.

Many scholars have studied flight delay issue based on different machine learning methods. Esmaeilzadeh and Mokhtarimousavi used a support vector machine to mine the nonlinear relationship between flight delay and various features. Given the black-box nature of machine learning, the sensitivity analysis of corresponding variables and

independent variables was conducted, and weather factors, airport scene operation, demand, and other factors were comprehensively considered. This research provided a new idea for studying the flight delay causes [3]. Kalyani et al. proposed a flight arrival delay prediction classification model based on XGBoost and a flight arrival delay prediction regression model based on linear regression. As one of the most widely used algorithms in the machine learning field, linear regression has the advantages of simple principle and easy application, and XGBoost is an ensemble learning algorithm based on the Decision Tree, which can find the optimal result by constantly adjusting the hyperparameters [4]. Zhang and Ma established a flight delay prediction model based on the Catboost algorithm, and the prediction accuracy reached 0.77. The SHAP value was used to analyze the features' contribution degree [5]. Khaksar and Sheikholeslami developed a hybrid method combining the J48 Decision Tree with *K*-means to train flight datasets from the United States and Iran, respectively, and compared them with four algorithms and obtained the optimal results with the hybrid method [6].

When utilizing machine learning techniques, most scholars will use multiple machine learning algorithms to train the same datasets and come up with the optimal algorithm and the optimal predict result through the evaluation indexes comparison [7, 8]. Moreover, with the development of machine learning technology, the variety of algorithms is increasing and most scholars tend to use at least three algorithms in one research. Henriques and Feiteira presented a classification model based on Hartsfield-Jackson International Airport which utilized Decision Tree, Random Forest, and Multilayer Perceptron. The Multilayer Perceptron provided the highest accuracy [9]. Choi et al. attempted two supervised learning algorithms, Decision Tree and KNN, and two ensemble learning algorithms, Random Forest, and Adaboost, and the results showed that ensemble algorithm classifier was greater than single algorithm classifier [10]. Stefanovič et al. took Lithuania Airport flight delays datasets as the research object and selected seven machine learning algorithms including probabilistic neural network, multilayer perceptron neural network, Gradient-Boosted Tree, Decision Tree, and the Gradient-Boosted Tree obtained the optimal results [11]. The above research studies are inspirational, and most of them through the model comparison obtain one optimal model while the other models were eliminated which create a waste of computing power. In addition, flight datasets are enormous and versatile, and the stability of algorithm is significant for real world applications. However, most studies did not pay attention to the algorithm stability, especially some novel algorithms. In thie study, we build a flight delay prediction classification model based on Stacking and design the experiments to verify the stability of Stacking.

The flight delay prediction methods based on machine learning technology become mature gradually. However, one core process that is often neglected in previous studies is feature selection [12]. Features selection is an essential step in machine learning [13]. The main purpose of feature selection is to remove redundant features and improve model efficiency by calculating feature importance. Onan and Korukoglu presented a feature selection model based on the ensemble method. The experiment result shows that the proposed method not only effectively processed the complex features but also improved the classification accuracy [14]. In addition, considering weather information could effectively improve the prediction accuracy [15], but the exact weather information might not be available until few hours before the flight. Therefore, we are not considering bringing in weather features in this research temporarily. The rest of this paper is organized as follows. Section 2 elaborates the research methods and principles used in this study including the Stacking classification algorithm, the SMOTE algorithm, the Boruta algorithm, and several indexes. Section 3 describes the data sources and the data preprocessing method. Section 4 discusses comparative experiments and comprehensively evaluates the prediction results through Accuracy, Precision, Recall, *F*1 Score, ROC curve, and AUC Score. In Section 5, the conclusions and expectations of this research are discussed.

2. Methodologies

2.1. Stacking Classification Methods. Stacking methods are derived from the idea of ensemble learning based on learners' combinations [16]. Stacking learner usually contains two levels, the first-level learner consists of multiple basics learners selected for training the same datasets, and the predicted outputs will become a new dataset to be carried into the second-level learner [17]. To avoid overfitting, cross-validation can be used when the first-level learner is the training model, and we select the *k*-fold cross-validation method in this paper [18]. The main process of Stacking methods is shown in Figure 1.

The initial datasets have been divided into training dataset *Dta* and testing dataset *Dts*, and then the training dataset *Dta* has been divided into *k* subdatasets, *Dta*1, *Dta*2,. . ., *Dtak*. In the *k*-fold cross-validation method, *i* models will be trained for *k* times, each subdataset becomes a test dataset in turn, and other subdatasets are training datasets to participate in training. In each model, *k* prediction results are combined to form a new training subdataset $Tir(r = 1,2, \ldots, k)$ and $Tir\ (r = 1,2, \ldots, k)$ have formed a new training datasets *Nta* and brought into the second-level learner.

When K-fold cross-validation is carried out in the first-level learner, every time Model *i* trains the training dataset *Dta,* testing datasets *Dts* will be predicted as well. Therefore, *k* prediction results *Rik* which are predicted by the same testing dataset *Dts* will be obtained. When solving the regression problem, the averaging method is usually adopted to process the *k* prediction results. In the classification problem, the processing of the prediction results is shown in Figure 2.

In machine learning, the binary classification will output the probability value of positive and negative at first. The category corresponding to a higher probability value is the category of the data sample, and the sum of the probability value is 1. In Stacking classification, model *i* predicts that the

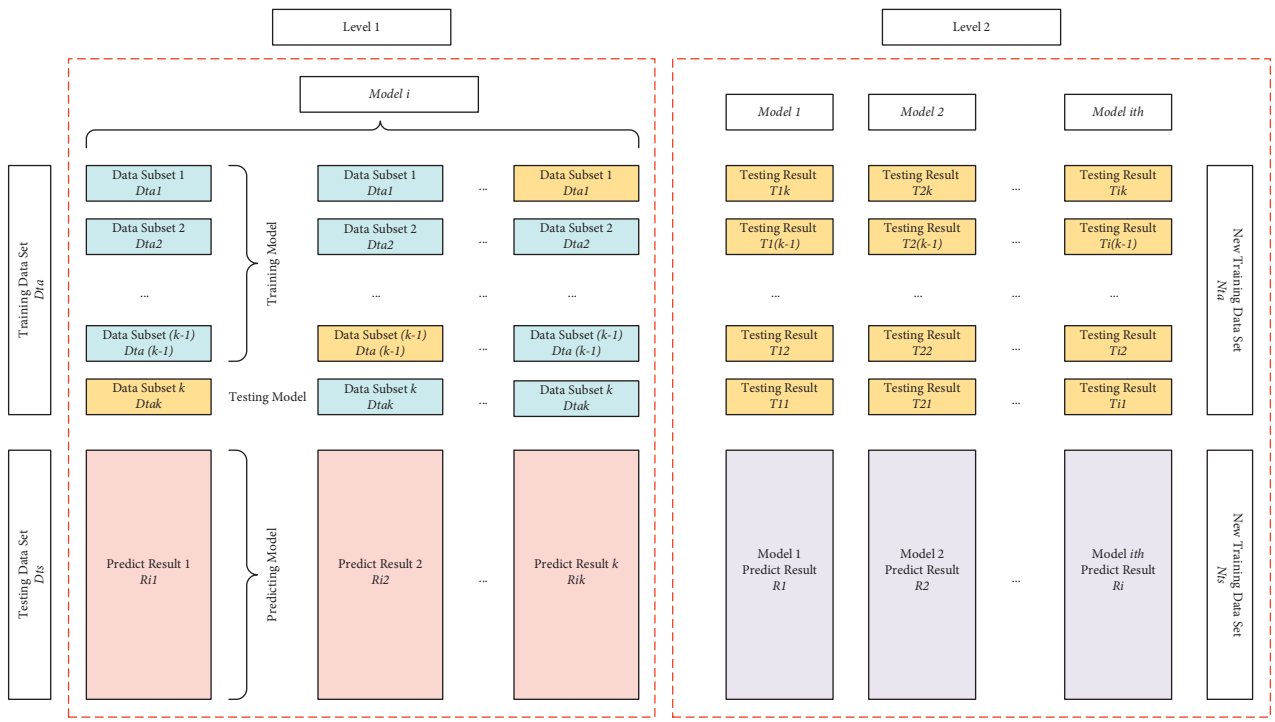


Figure 1: Stacking methods framework.

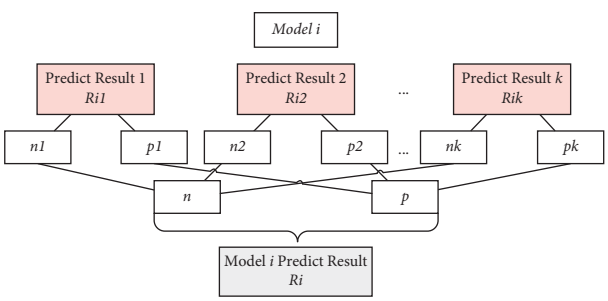


Figure 2: Stacking classification method of the second-level learner framework.

probability of the data sample belonging to positive p is $P(p) = (p1 + p2 + \cdots + pk)/k$ and the probability of the data sample belonging to negative is $P(n) = (n1 + n2 + \cdots + nk)/k$. Thus, the prediction result of Model i on testing dataset Dts, Ri ($i = 1,2, \ldots, i$), forms a new testing dataset Nts into the second-level learner. The second-level learner could choose a relatively simple algorithm and then trains the model with the new training dataset Nta and test with new testing datasets Nts.

2.2. Imbalanced Datasets Processing. Imbalanced datasets are one of the common problems in machine learning classification. This is mainly reflected in the fact that the number of samples belonging to a certain category in the datasets is far greater than that of other categories. To improve the accuracy, most classification algorithms tend to identify the minority class data samples as the majority class samples when training imbalanced datasets. Although such a classifier can achieve a certain accuracy, it does not have applicability [19]. The flight delay datasets in this paper are typical imbalanced datasets, and the data volume of on-time flights is nearly four times that of delayed flights (3.78 : 1).

Oversampling and undersampling are the commonly used techniques to deal with imbalanced datasets [20]. The main idea of these two technologies is to reconstruct the sample size. Undersampling has achieved balance by reducing most samples, while Oversampling has achieved balance by increasing the minority of samples.

In this paper, SMOTE (synthetic minority oversampling technique) algorithm is selected to process the imbalanced datasets [21]. The SMOTE algorithm is an oversampling technology based on the KNN algorithm. It improves the simple random oversampling algorithm of randomly copying a few samples to increase the sample size, which can avoid overfitting and effectively improve the generalization ability of the model. The main process of the SMOTE algorithm is as follows:

(1) The Euclidean distance is calculated from each minority sample x to the other minority sample

(2) The sampling rate is set according to the difference between the minority sample size and the majority sample size and randomly determines k nearest neighbors of sample x of a minority class

(3) Between a few samples x and x_i, according to the sampling rate set in Step (2), a new sample x_n can be calculated according to the following formula:

$$x_n = x + ran\ d(0,1) \times |x - x_i|. \quad (1)$$

2.3. Features Selection. Feature selection is one of the core contents of machine learning, which aims to eliminate redundant features, improve model accuracy, and reduce operation time. The commonly used feature selection methods include Filter, Wrapper, and Embedded [22]. The Boruta algorithm is utilized in this research to select features. Boruta is an encapsulated feature selection algorithm based on Random Forest. The importance of each feature to the dependent variable is calculated to determine whether to be retained. The main process of the Boruta algorithm is as follows:

(1) Establish shadow feature: the original features are randomly sorted to form a shadow feature matrix, and the new feature matrix is obtained by splicing the shadow feature matrix with the original feature matrix.

(2) The new feature matrix is brought in a Random Forest classifier for training, and output the importances of features v.

(3) The Z score of the original feature and shadow feature is calculated, and the calculation formula is as follows:

$$z_{\text{score}} = \frac{A_v}{S_v}, \quad (2)$$

where $A_v$represents the average value of feature importance and $S_v$represents the standard deviation of feature importance.

(4) The maximum z_{score} is searched in the shadow feature, denoted as Z_{max}.

(5) If the original feature z_{score} is greater than Z_{max}, the feature is recorded as "important." On the contrary, if the original feature z_{score} is less than Z_{max}, the feature will be marked as "unimportant" and be deleted.

(6) Steps (1) to (5) are repeated until all features have been marked.

2.4. Evaluation Indexes. In this paper, Accuracy, Precision, Recall, and $F1$ Score are calculated by output confusion matrix to evaluate the prediction results. The confusion matrix is shown in Figure 3 [23].

TP is True Positive, indicating that both the true value and the predicted value are positive, that is, the number of positive samples predicted correctly. FP is False Positive, indicating that the true value is negative, but the predicted value is positive, that is, the number of negative samples is wrongly predicted to be positive. TN is True Negative, indicating that both the true value and the predicted value are negative, that is, the number of negative samples that are correctly predicted. FN is False Negative, indicating that the true value is positive, but the predicted value is negative, that is, the number of positive samples that are wrongly predicted to be negative.

Confusion Matrix		Predicted	
		Positive	Negative
True	Positive	*TP*	*FN*
	Negative	*FP*	*TN*

Figure 3: Confusion matrix.

Accuracy is the ratio of correctly predicted samples to the total amount of samples, and its calculation formula is as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \cdot 100\%. \quad (3)$$

Accuracy is one of the most used evaluation indexes in classification. Since the flight delay data sample is the imbalanced dataset, that is, the sample size of on-time flights is much larger than delayed flights. To improve accuracy, the model tends to identify the minority samples as the majority, and the model can obtain higher accuracy, but the prediction of delayed samples is almost ineffective. Therefore, the predicted results also need to be evaluated by Precision, Recall, and $F1$ Score in the classification problem.

Precision indicates the percentage of correct predictions in the sample with a positive predicted value. The calculation formula is as follows:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \cdot 100\%. \quad (4)$$

Recall indicates the percentage of the correct prediction in the sample with a positive true value. The calculation formula is as follows:

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \cdot 100\%. \quad (5)$$

According to the calculation formula of Precision and Recall, it can be found that when the Precision increases, the Recall will decrease, and when the Recall increases, the Precision will decrease. In this paper, the Precision focuses on how many delayed flights were successfully predicted in the total sample, while the Recall focuses on how many delayed flights were successfully predicted in all delayed flights. Moreover, the $F1$ Score, as the harmonic average of Precision and Recall, could consider both. The calculation formula is as follows:

$$F1\ \text{score} = \frac{2 \cdot \text{Precisiong} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (6)$$

3. Data Acquisition and Preprocessing

3.1. Data Sources. In this research, we collect flight data from January to December 2019 at Logan International Airport in Boston, Massachusetts, the United States. The total number of departure flight datasets is 149,576, and the total number of arrival flight datasets is 149338. The Logan Airport is one

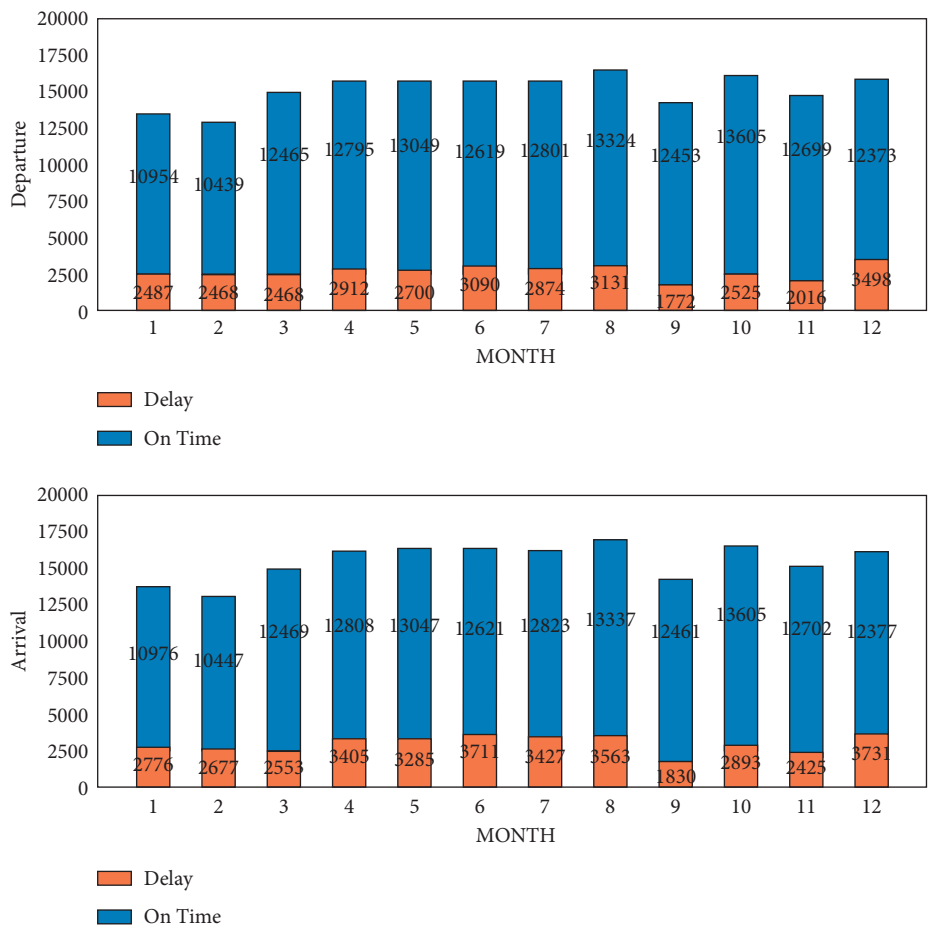


Figure 4: Monthly number of delayed flights and on-time flights.

of the busiest airports in the eastern United States, with 31, 941 flights delayed in the departure dataset and 35,941 flights delayed in the arrival dataset. The departure delay rate is 21.35%, and the arrival delay rate is 24.07%. The monthly distribution of flight delays in 2019 is shown in Figure 4.

Both datasets include 9 features, and the input features and descriptions are shown in Table 1.

3.2. Uniformization Processing. To avoid the impact of dimensionless differences among features in the dataset, the data are normalized in this paper. The aim is to adjust the mean of the data to 1 and the variance to 0. The calculation formula is as follows:

$$x' = \frac{X - X_{\text{mean}}}{X_{\max} - X_{\min}}, \tag{7}$$

where X_{mean} is the mean value, $X_{\max}$ is the maximum value, and $X_{\min}$ is the minimum value.

4. Experiment and Analysis

4.1. Features Selection Results. In this research, the Boruta algorithm is utilized to select features for the departure delay dataset and arrival dataset, respectively, and the results are shown in Figure 5. In the departure dataset, all features are marked as important. The CRS_DEP_TIME is the most important feature in the departure dataset. In the arrival dataset, 8 features are estimated as important features, and Diverted has been rejected. The departure dataset features importance is shown in Table 2, and the arrival dataset features importance is shown in Table 3.

Table 1: The input features and descriptions.

Features	Format	Description
Quarter	int64	Quarter (1–4)
Month	int64	Month (1–12)
Day_of_month	int64	Day of month (1–31)
Day_of_week	int64	Day of week (1–7)
CRS_dep_time	int64	CRS departure time (local time: hhmm)
CRS_arr_time	int64	CRS arrival time (local time: hhmm)
CRS_elapsed_time	int64	CRS elapsed time, in minutes
Distance	int64	Miles
Diverted	int64	diverted = 1, not diverted = 0

To explore the influence of features' importance on the prediction results, the following experiment has proceeded. At first, only input the most important features for training and then add one feature at a time according to the importance value until all the features are input. According to

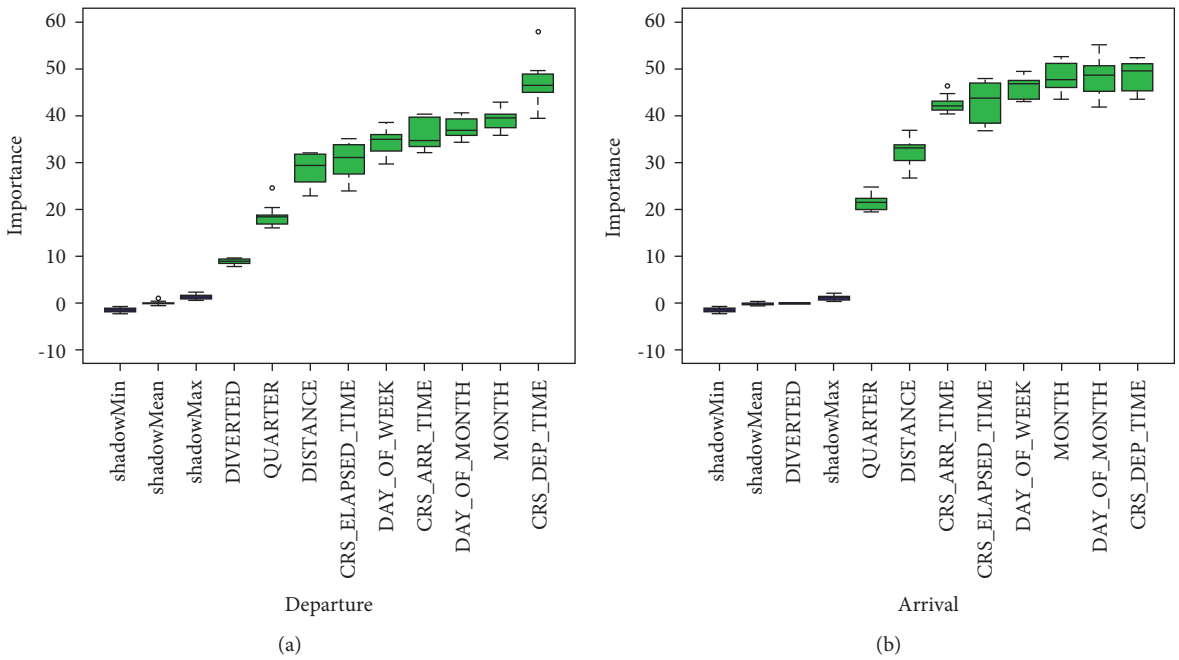


Figure 5: Features selection results: (a) departure; (b) arrival.

Table 2: Departure dataset features importance.

	meanImp	medianImp	minImp	maxImp	normHits	Decision
Quarter	18.7022	18.38765	15.96021	24.59658	1	Confirmed
Month	39.15992	39.53052	35.61664	42.88205	1	Confirmed
Day_of_month	37.13472	36.92707	34.23672	40.63482	1	Confirmed
Day_of_week	34.41572	34.80809	29.75517	38.51781	1	Confirmed
CRS_dep_time	47.17696	46.49994	39.36456	57.88962	1	Confirmed
CRS_arr_time	35.90235	34.88594	32.14023	40.42103	1	Confirmed
CRS_elapsed_time	30.42818	31.15777	24.04919	35.14513	1	Confirmed
Diverted	8.957969	9.035397	7.912347	9.665486	1	Confirmed
Distance	28.66349	29.39897	22.9603	31.81813	1	Confirmed

Table 3: Arrival dataset features importance.

	meanImp	medianImp	minImp	maxImp	normHits	Decision
Quarter	21.59	21.61953	19.64103	24.85327	1	Confirmed
Month	48.19586	47.78515	43.47335	52.73112	1	Confirmed
Day_of_month	48.35522	48.61903	41.88918	55.16763	1	Confirmed
Day_of_week	46.23737	46.70293	43.08921	49.43334	1	Confirmed
CRS_dep_time	48.68234	49.67468	43.40899	52.34451	1	Confirmed
CRS_arr_time	42.41878	42.13526	40.19638	46.36798	1	Confirmed
CRS_elapsed_time	42.78774	43.67602	36.73169	47.83175	1	Confirmed
Distance	0	0	0	0	0	Rejected
Diverted	32.28774	33.28389	26.69778	36.87651	1	Confirmed

the feature selection results, the Diverted is removed in arrival prediction model training. In this experiment, the first-level learner contains five algorithms: Decision Tree, KNN, Logistic Regression, Gaussian Naive Bayes, and Random Forest. The second-level learner is Logistic Regression. The experiment results are shown in Figure 6.

In the departure dataset, when the fifth important feature is given as input, Accuracy, Precision, Recall, and $F1$ Score exceed 0.8. When the sixth important feature is given as input, the indexes show slight decrease, but the overall trend is stable without significant increase or decrease. In other words, the last four features contributed limited to the prediction model, which was consistent with Boruta feature selection results. In the arrival dataset, when the fourth important feature is given as input, the evaluation indexes have no significant change. In the

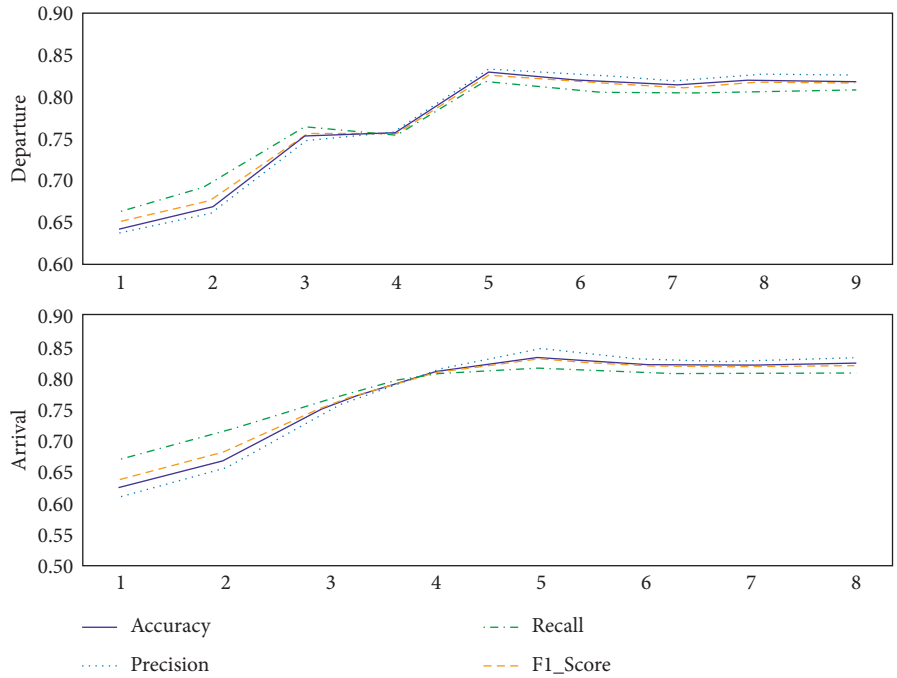


Figure 6: The prediction results with different features.

arrival dataset, when the fourth feature is given as input, the evaluation indexes exceed 0.8 and tend to be stable. It is worth mentioning that with the increase in features, Recall changes from the highest to the lowest among the four indexes, while Precision changes from the lowest to the highest.

4.2. Comparison between Algorithms. There is no "multipurpose algorithm" or "the greatest algorithm" in machine learning. It is necessary to attempt multiple algorithms. In this research, six algorithms are selected including KNN, Random Forest, Logistic Regression, Decision Tree, Gaussian Naive Bayes, and Stacking to train the same dataset, respectively. The experiment results are shown in Figure 7. In addition to Stacking, Random Forest also showed a great prediction result which four evaluation indexes all exceed 0.8. The difference among four indexes of KNN is lager than other algorithms but also has reached 0.7. Meanwhile, Gaussian Naive Bayes and Logistic Regression have relatively poor performance, and four indexes are around 0.6.

The ROC (receiver operating characteristic) curve could measure algorithm generalization ability. The AUC (area under curve) is the area under the ROC curve [24]. The closer the AUC is to 1, the better the algorithm will be. We output the ROC for each algorithm and calculate the AUC Score, and the results are shown in Figure 8. Stacking reaches 0.823 in the departure dataset and 0.821 in the arrival dataset. The result of Random Forest is similar to that of Stacking. With this result, we consider that Random Forest contributes more to Stacking compared with other algorithms. However, if we remove Random Forest from the Stacking algorithm, will the performance of Stacking decrease? In other words, if we remove the weak performance algorithm Gaussian Naive Bayes, will the performance of Stacking increase? In section 4.3, we experiment to explore the impact of strong and weak algorithms on the performance of Stacking.

4.3. First-Level Learners Analyses. In the single algorithm comparison, we find that the Random Forest has great performance, and Gaussian Naive Bayes and Logistic Regression perform poorly. In this section, one algorithm is removed, in turn, to figure out how strong or weak algorithms affect Stacking prediction results. The results are shown in Tables 4 and 5. Overall, there is no significant difference between the six groups with different first-level learners. Both in the departure dataset and arrival dataset, the four evaluation indexes are similar among the six scenarios, only the Recall and $F1$ Score of the third scenario decrease below 0.8. The overall accuracy is shown in Figure 9. The prediction accuracy is around 0.8 which is close to the result of Stacking. It can be concluded that the Stacking algorithm not only could ensure the prediction accuracy but also maintains great stability. Random Forest has a strong performance, but when we remove Random Forest form the first-level learner, the model still acquires great predict results. As we mentioned before, there is no "multipurpose algorithm" or "the greatest algorithm" in machine learning. Therefore, the Stacking algorithm could be a great solution to deal with algorithm selection, especially the enormous and complex datasets like flight datasets.

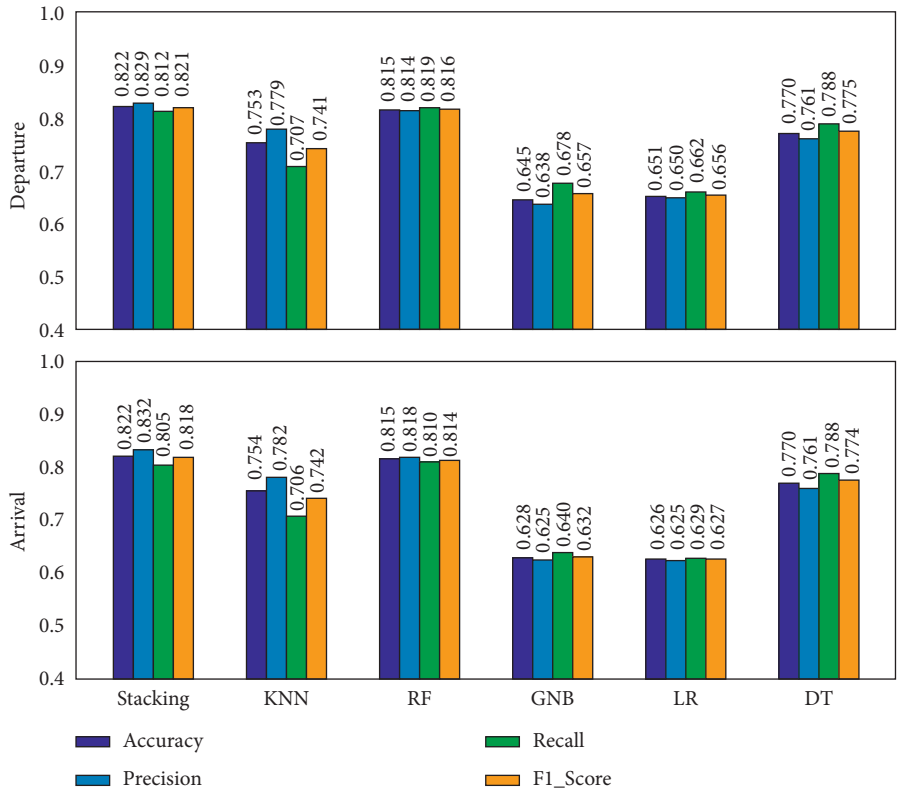


Figure 7: The prediction results of different algorithms.

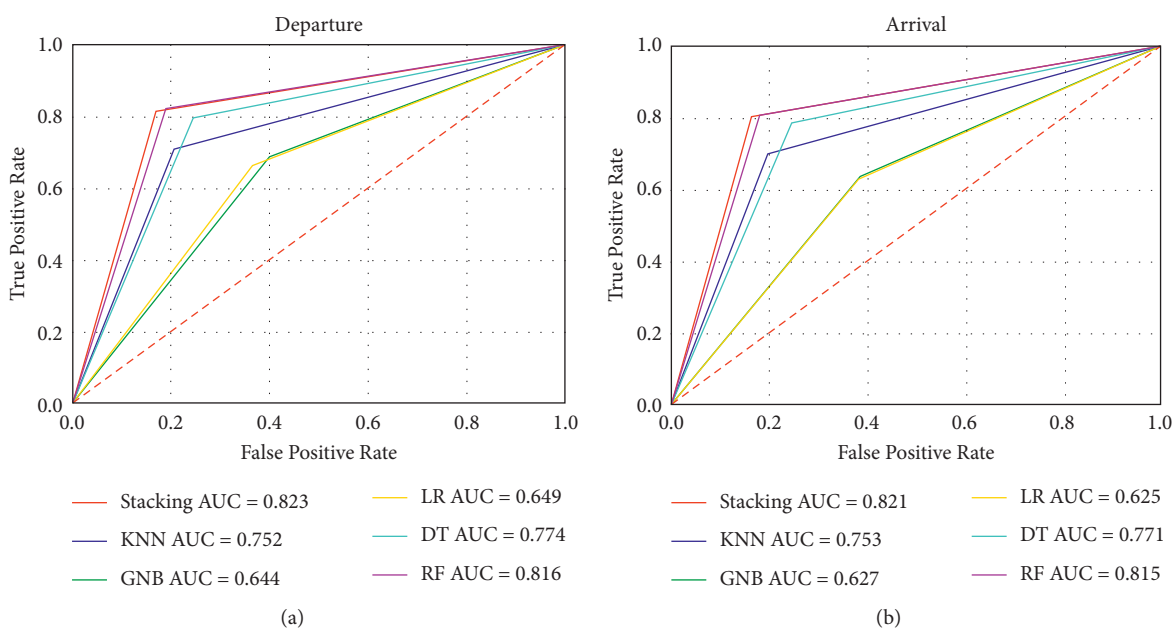


Figure 8: Receiver operating characteristic curve: (a) departure; (b) arrival.

Table 4: The departure prediction results of different first-level learners.

Departure	First-level learner	Accuracy	Precision	Recall	*F*1 Score
1	GNB, RF, KNN, LR, DT	0.822	0.830	0.812	0.821
2	RF, KNN, LR, DT	0.821	0.8277	0.812	0.820
3	GNB, KNN, LR, DT	0.800	0.805	0.784	0.794
4	GNB, RF, LR, DT	0.819	0.823	0.812	0.817
5	GNB, RF, KNN, DT	0.822	0.828	0.811	0.819
6	GNB, RF, KNN, LR	0.82	0.827	0.811	0.819

Table 5: The arrival departure prediction results of different first-level learners.

Arrival	First-level learner	Accuracy	Precision	Recall	*F1* Score
1	GNB, RF, KNN, LR, DT	0.822	0.832	0.808	0.82
2	RF, KNN, LR, DT	0.82	0.827	0.806	0.816
3	GNB, KNN, LR, DT	0.80	0.811	0.78	0.793
4	GNB, RF, LR, DT	0.818	0.824	0.804	0.814
5	GNB, RF, KNN, DT	0.82	0.828	0.805	0.816
6	GNB, RF, KNN, LR	0.818	0.825	0.804	0.814

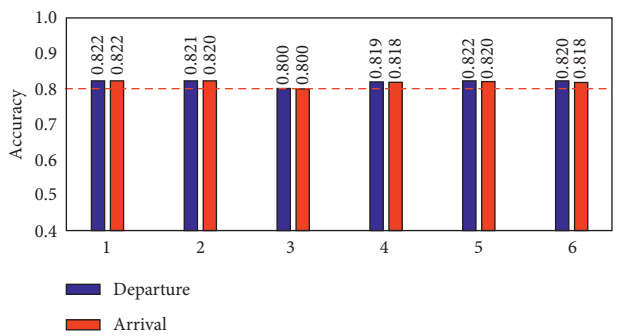


Figure 9: The accuracy of different first-level learners.

5. Conclusion

In this research, we propose a flight delay prediction classification method based on the Stacking algorithm. The SMOTE algorithm is introduced to process imbalanced datasets used, and the Boruta algorithm is utilized to select input features. The Logan International Airport flight data in 2019 are collected to carry out comparative experiments, and the Accuracy, Precision, Recall, and *F*1 Score are above 0.8. The main contributions are as follows:

(1) The Boruta algorithm is used to select features. Features selection is an essential process when utilizing machine learning technology. According to section 4.1, the comparison experimental results are consistent with the Boruta algorithm feature selection results, which verify the effectiveness of the Boruta algorithm. 9 feature importances are obtained based on the Random Forest classifier, and the experiments are designed to input different features into the model in the order of their importance value. In the departure dataset, all features have been confirmed while Diverted has been rejected in the arrival dataset.

(2) A flight delay prediction classification method based on Stacking is proposed in this study. The first-level learner includes KNN, Random Forest, Logistic Regression, Decision Tree, and Gaussian Naive Bayes, and the second-level learner utilizes Logistic Regression. To distinguish the contribution of five first-level learners, the same dataset that has been trained based on these five first-level learners separately. The result shows that Random Forest has the best performance which is similar to Stacking.

(3) The main aim of this study is to explore the stability of the Stacking algorithm. Stacking is a combination of different algorithms with different performances. In section 4.3, we design an experiment to verify how strong or weak learners affect the Stacking performance. The experiment result shows that whether strong learners or weak learners are removed, the overall accuracy of the Stacking has no obvious difference. Therefore, we believe that Stacking provides a reliable solution for algorithm selection in machine learning applications, especially the enormous and complex datasets like flight datasets.

In future research, other machine learning technologies can be utilized to study flight delay prediction. Moreover, it can also pay close attention to weather influence on a flight delay. In this research, we does not add exact weather-related features in the prediction model but that does not mean weather influence is unimportant. On the contrary, we believe that studying the influence of weather on flight delays is a significant and complex issue. We will focus more on establishing reasonable features to measure the impact of weather on flight delays, especially for high-impact weather, and use machine learning correlation analysis technology to explore the relatedness between weather and flight delay.

Data Availability

The flight dataset used in this paper is from the Bureau of Transportation Statistics website (https://www.transtats.bts.gov/homepage.asp).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2018YFE0208700) and the National Natural Science Foundation of China (No. 52002177).

References

[1] Bureau of Transportation Statistics, "Bureau of Transportation Statistics,".
[2] M. Ball, C. Barnhart, M. Dresner et al., "Total delay impact study," 2010.

[3] E. Esmaeilzadeh and S. Mokhtarimousavi, "Machine learning approach for flight departure delay prediction and analysis," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 8, pp. 145–159, 2020.

[4] N. L. Kalyani, G. Jeshmitha, U. Bindu Sri Sai, M. Samanvitha, J. Mahesh, and B. V. Kiranmayee, "Machine learning model - based prediction of flight delay," in *Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, November 2020.

[5] B. Zhang and D. Ma, "Flight delay prediciton at an airport using maching learning," in *Proceedings of the 2020 5th International Conference on Electromechanical Control Technology and Transportation*, Nanchang, China, May 2020.

[6] H. Khaksar and A. Sheikholeslami, "Airline delay prediction by machine learning algorithms," *Scientia Iranica*, vol. 1, p. 12, 2017.

[7] G. Rebala, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning*, Springer International Publishing, New York, NY, USA, 2019.

[8] A. Onan, "On the performance of ensemble learning for automated diagnosis of breast cancer," *Advances in Intelligent Systems and Computing*, vol. 347, pp. 119–129, 2015.

[9] R. Henriques and I. Feiteira, "Predictive modelling: flight delays and associated factors, hartsfield-jackson atlanta international airport," *Procedia Computer Science*, vol. 138, pp. 638–645, 2018.

[10] S. Choi, Y. J. Kim, B. Simon, and D. Mavris, "Prediction of weather-induced airline delays based on machine learning algorithms," in *Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, CA, USA, December 2016.

[11] P. Stefanovič, R. Štrimaitis, and O. Kurasova, "Prediction of flight time deviation for Lithuanian airports using supervised machine learning model," *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 8878681, 10 pages, 2020.

[12] G. Gui, F. Liu, J. Sun, J. Yang, Z. Zhou, and D. Zhao, "Flight delay prediction based on aviation big data and machine learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 140–150, 2020.

[13] N. Chakrabarty, "A data mining approach to flight arrival delay prediction for american airlines," 2019, https://arxiv.org/abs/1903.06740.

[14] A. Onan and S. Korukoglu, "A feature selection model based on genetic rank aggregation for text sentiment classification," *Journal of Information Science*, vol. 99, pp. 1103–1107, 2015.

[15] Y. J. Kim, C. Sun, S. Briceno, and D. Mavris, "A deep learning approach to flight delay prediction," in *Proceedings of the Digital Avionics Systems Conference 2016*, Sacramento, CA, USA, December 2016.

[16] Z. H. Zhou, *Ensemble Methods: Foundations and Algorithms*, Taylor & Francis, Oxfordshire, UK, 2012.

[17] G. Zhong, T. Yin, L. Li, J. Zhang, H. Zhang, and B. Ran, "IEEE intelligent transportation systems magazine," *IEEE Intelligent Transportation Systems Magazine*, vol. 99, 2020.

[18] J. D. Rodriguez, A. Perez, and J. A. Lozano, "Sensitivity analysis of *K*-fold cross validation in prediction error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 569–575, 2010.

[19] H. Patel, D. S. Rajput, G. T. Reddy, C. Iwendi, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 16, no. 4, Article ID 812444408, 2020.

[20] A. Behzad Mirzaei, A. B. Bahareh Nikpour, and A. Nezamabadi Pour, "A clustering and density-based hybrid approach for imbalanced data classification," *Expert Systems with Applications*, vol. 164, 2020.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[22] Q. Al-Tashi, H. Md Rais, S. Mirjalili, and H. Alhussian, *A Review of Grey Wolf Optimizer-Based Feature Selection Methods for Classification*, UTP Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia, 2020.

[23] S. A. Alvarez, *An Exact Analytical Relation Among Recall, Precision, and Classification Accuracy in Information Retrieval*, Boston College, Newton, MA, USA, 2002.

[24] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2005.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 3584569, 16 pages
https://doi.org/10.1155/2021/3584569

Review Article

Traffic Signal Optimization under Connected-Vehicle Environment: An Overview

Jindong Wang,[1,2] Shengchuan Jiang,[1] Yue Qiu,[1] Yang Zhang,[3] Jianguo Ying,[2] and Yuchuan Du[1]

[1]*Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai, China*
[2]*Jinqiao (Group) Co., Ltd., Shanghai, China*
[3]*Shanghai Urban-Rural Construction and Transportation Development Institute, Shanghai, China*

Correspondence should be addressed to Shengchuan Jiang; shengchuanjiang@tongji.edu.cn

Received 10 May 2021; Revised 19 July 2021; Accepted 3 August 2021; Published 11 August 2021

Academic Editor: Chi-Hua Chen

Traffic signal optimization is a significant means for smoothing urban traffic flow. However, the operation of traffic signals is currently seriously constrained by the data available from traditional point detectors. In recent years, an emerging technology, connected vehicle (CV), which can percept the overall traffic environment in real time, has drawn researchers' attention. With the new data source, traffic controllers should be able to make smarter decisions. A lot of work has been done to develop a new traffic signal control pattern under connected-vehicle environment. This paper provides a comprehensive review of these studies, aiming at sketching out the state of the arts in this research field. Several basic control problems, communication, control input, and objectives, are briefly introduced. The commonly used optimization models for this problem are summarized into three types: rule-based models, mathematical programming-based models, and artificial intelligence-based models. Then some major technical issues are discussed in detail. Finally, we raise the limitation of the existing studies and give our perspectives of the future research directions.

1. Introduction

The recent years have witnessed an exponential increase in vehicular traffic in urban areas. As a result, a variety of challenges have emerged, including traffic congestion, energy consumption, traffic safety risk, and pollutant emissions. As reported by the Texas A&M Transportation Institute and INRIX, Americans spent 6.9 billion hours of extra time and purchased 3.1 billion gallons of extra fuel due to congestion in 2014, causing average cost of $960 for every autocommuter compared to an inflation-adjusted $400 in 1982 [1]. In another report, the total number of crashes in 2015 was over 6.2 million, 3.8% higher than that in the previous year [2]. For the same period, the ratio of total carbon dioxide (CO_2) emissions attributable to transportation reached 32.1%, about 3% higher than the ratio in the year of 1990 [3]. With the rapid motorization, especially in developing countries, it is expected that 70% of the people worldwide will live in cities [4] and the number of operating vehicles around the world will at least double by 2050 [5]. Unless effective measures are taken, these problems, therefore, may deteriorate in the future.

To solve the aforementioned problems, one may consider expanding the road facilities to satisfy the increasing travel demand. But a more feasible and low-cost solution is to optimize the traffic signal to balance the traffic load and decrease waiting times, based on the fact that road intersection is the bottleneck of the traffic network and thus one of the major contributors to traffic inefficiency. Since the first prototype colored traffic signal light was installed in England in 1868 [6], signal control systems have experienced ongoing development. Nowadays, the in-use traffic control strategies can be categorized into three types: (1) fixed-time, (2) actuated, and (3) adaptive control.

(1) Fixed-time control: Fixed-time control predefines a static and repeating sequence of phases and durations based on historical data to serve different times of the day (TOD) to address the demand fluctuation. For the reason that the timing plan is designed based on historical data in an offline manner, this control strategy is also known as offline control. The underlying assumption of fixed-time control is that the traffic demand remains unchanged within the entire time period of a timing plan. Examples include Webster's method [6] and its extensions, SIGSET [7], SIGCAP [8] for isolated control together with MAXBAND [9] and its extensions, TRANSYT [10], and MULTIBAND [11] for coordinated control.

(2) Actuated control: Actuated control detects the dynamic traffic demand to modify a fixed timing plan by occasionally skipping a phase if no vehicle is present or shortening a phase when vehicles are not being served. Actuated control and the following adaptive control enable traffic signal controllers to make use of real-time traffic information (e.g., vehicle counts, lane volume, and lane occupancy), so the two control methods belong to online control. MOVA [12] is a typical actuated control system which uses the traffic data upstream of the stop-line to implement control logic. In the USA, actuated signal control is widely adopted according to the signal control guidelines recommended by the Federal Highway Administration [13].

(3) Adaptive control: Adaptive timing plans attempt to continuously change their signal phases and timings in response to real-time traffic conditions. Therefore, this method is more flexible than actuated control. Adaptive control was first proposed as early as 1960s [14]. The early studies and implementations include SCOOT [15], SCATS [16], PRODYN [17], OPAC [18], and UTOPIA [19]. The recently developed systems such as RHODES [20] can realize proactive control by predicting traffic demands at a downstream intersection and optimizing lost times on a global scale.

The advantage of fixed-time control systems lies in calculation simplicity as well as lower hardware requirements. But the pretimed timing plans cannot capture occasional events such as traffic accidents and road closure. In addition, the variation trend of traffic volume is not exactly identical on different days. Therefore, actuated and adaptive control systems are increasingly adopted in metropolises in order to deal with the downsides of fixed-time control. However, there are still several unsolved problems. All the aforementioned actuated and adaptive signal control systems collect real-time traffic data from infrastructure-based sensors, for example, loop detectors, ultrasonic detectors, or video detectors, which can only conduct point detection and estimate traffic states based on very limited information, such as vehicle counts or temporal gap between consecutive vehicles. Inductive loop is broken down frequently in practical application, while the ultrasonic sensor and video detector are very sensitive to the weather, resulting in degraded system performance. Moreover, installation and maintenance of these detectors require frequent and costly investments. In summary, the existing systems are not able to collect traffic information comprehensively, steadily, and at low cost.

In this context, Intelligent Transportation Systems (ITS) are considered as the key for enhancing the capacity of the traffic control systems. With the advances in wireless communication technology, vehicular ad hoc networks (VANETs) are created by applying the principles of mobile ad hoc networks (MANETs) to the domain of vehicles and become a significant part of ITS framework [21]. In VANETs, vehicles are able to communicate with each other (vehicle-to-vehicle communication, V2V) and with the infrastructure (vehicle-to-infrastructure communication, V2I) through dedicated short-range communications (DSRC) (or any other wireless networking technology) and are referred to as connected vehicles (CVs). Data from CVs provide a complete picture of the vehicle states including location, speed, acceleration, and other vehicle data. Compared with the conventional point detection sensors, the traffic information provided by CVs is also steadier and more persistent because fault of an individual vehicle will only slightly decrease the penetration rate and the system can still provide relatively accurate information. In addition, such a system saves installation and maintenance cost of specific traffic flow detectors. Based on the new source of data, traffic controllers should be able to make smarter decisions. In the last decade, researchers began to explore the benefits of using such information. For example, Gradinescu et al. [22] designed an adaptive traffic light system relying on wireless communication between vehicles and traffic light controller, where the algorithm is still based on the traditional Webster's signal timing formula but all the information needed is collected by VANETs; Kari et al. [23] proposed an online adaptive traffic signal control (ATSC) strategy based on CV technology, which is capable of adjusting traffic light settings, including green splits and phase sequence in response to the variations in traffic demand and arrival pattern. The studies show the potential of CV technology used in traffic signal optimization.

Another trend, as is well known, is the development of autonomous driving technology. The connected vehicles combined with autonomous driving technology, so-called connected and autonomous vehicles (CAVs), is being hailed by both academia and industry as the revolution of human mobility. The National Highway Traffic Safety Administration (NHTSA) defined vehicle automation into five levels, ranging from vehicles that do not have any automated control functions (level 0) through fully automated vehicles (level 4) [24]. In a long time of the future, vehicles with different technical levels may coexist in the transportation systems. So how to improve traffic operations in a road environment containing both traditional vehicles and autonomous vehicles is a key problem referring to traffic

control in the near future. On the other hand, vehicle type can be easily identified through the information provided by connected vehicles. Hence, traffic signal priority (TSP) can be implemented in new ways and, therefore, is equally worth of focusing on nowadays.

The major contribution of the presented paper is to give a comprehensive review of the state-of-the-art techniques with respect to traffic signal optimization under connected-vehicle environment. It should be noted that traffic control generally includes controlling the traffic at intersections, ramp-metering, and variable speed limits as well as route guidance, but the scope of the review is limited in intersection signal control. Information about ramp-metering under CV environment is available in other literatures such as [25, 26]. In the remainder of the paper, we first introduce some fundamental problems referring to V2X-based traffic signal control including communication systems, data requests, and control objectives. Then optimization models are reviewed according to the model paradigms. The following is the discussion about some major technical issues. Although immense amounts of concrete research focused on this topic, there are still many common problems which remain to be solved. In the last section, we will raise several remaining problems and state our viewpoints about the future research directions.

2. V2X-Based Traffic Signal Control System

2.1. Basics of Vehicular Communication. CV communication is an emerging technological framework that aims at direct data transmission between vehicles to vehicles (V2V) and vehicles to road infrastructure (V2I) using wireless technology. It is generally known that CV communication standards consist of three components: (1) IEEE 1609 "Family of Trial-Use Standards for Wireless Access in Vehicular Environments (WAVE)" [27], (2) IEEE 802.11p "Standard for Information Technology" [28], and (3) Society of Automotive Engineers International (SAE) J2735 "DSRC Message Set Dictionary" [29]. These standards are sometimes called WAVE/DSRC for short. The IEEE 1609 family defines an overall structure of the WAVE interface. IEEE 802.11p deals with the multichannel operations of the MAC layer. SAE J2735 defines the framework of DSRC messages, for example, here-I-am (HIA), a-la-carte (ALC), and signal phase and timing (SPaT) messages, to ensure interoperability among any possible CV applications. For safety applications, SAE J2735 also defines the Basic Safety Message (BSM), which is the most fundamental building block that enables proximity awareness. Such framework includes the elements and the usages of possible messages categorized based on the types of applications.

The unique difference of WAVE/DSRC as compared to existing wireless communication standards is that it does not require an authentication procedure. That is, under existing wireless LAN protocols, like IEEE 802.11a/b/g, a mobile node (i.e., a laptop or a smart phone) must be identified by an access point (AP) to join the network. However, these identification steps normally take a few seconds or even minutes and are thus not suitable for a mobile network composed of fast-moving vehicles. Thus, by omitting such identification steps, WAVE/DSRC enables quick connections between transceivers.

The channel operation of WAVE/DSRC is also of interest. The DSRC has a 75 MHz frequency spectrum at 5.9 GHz frequency band. The spectrum is divided every 10 MHz, resulting in seven different channels from 172 to 184 [28]. Channel 178 is called a control channel (CCH), and the other channels are called service channels (SCHs), except for the two channels at both ends that are reserved for future use. While the CCH is dedicated to the transmission of control messages such as beaconing or urgent safety-related messages, the SCHs are designed for exchanging any data packets including vehicular mobility information or commercial services. Therefore, the control messages or safety-related data that are transmitted through the CCH are not affected by data transmission through a SCH, thereby enabling the system to be suitable for the fast-moving mobile network.

On the basis of the above technologies, each vehicle transmits its temporary ID, location, speed, heading, lateral and longitudinal acceleration, brake system status, and vehicle size to surrounding vehicles and the infrastructure. By "listening" to these messages, a signal controller could gain a more comprehensive understanding of the movements of nearby vehicles than with traditional point detectors (e.g., loop detectors). Substantial improvement was seen in several real-world pilot projects such as Audi [30] and BMW [31].

2.2. Data Requirement of Traffic Signal Control. Traffic signal optimization algorithms need appropriate input including both static information about road facilities and real-time dynamic traffic states. Static information is usually preset in optimization models, while dynamic information depends on real-time communication between vehicles to vehicles and vehicles to infrastructures. In the reviewed literatures, the used real-time traffic information can be categorized into aggregative and individual vehicle information. The advantage of aggregative information lies in modest resource requirements such as average transmission content and computation power. In contrast, the upside of individual information is fine-grained considerations of the optimization problem. Examples of aggregative information include total vehicle counts, equipped vehicle ratio [32], and queue length in the lanes related to each traffic lights phase [33, 34]. Required individual vehicle information includes real-time location [35], speed [36], heading [37], and acceleration [38] as well as vehicle type [39] of each vehicle within a certain spatial range. Sometimes, weather and minor events [40] are also taken into account to improve the operations.

2.3. Control Objectives. In the traffic signal timing optimization process, one or more of the measures of effectiveness (MOEs) are optimized under certain constraints to determine appropriate signal timing parameters including phase plans, cycle lengths, green splits, and offsets. Different

objective functions and their combinations are used to define the problem. Therefore, proper selection of the objective function for signal timing optimization is a very important task. Minimizing time lost in the intersections, such as average delay, average waiting time, or travel time, is the most commonly selected objective function [41–43]. Some of the studies also considered the variance of waiting time to improve fairness [44]. Several studies conducted queue-based optimization, that is, minimizing average or maximum queue length [34, 45]. In order to consider the environmental influence, decreasing pollutant emissions can be imported as an objective function [46]. It is noted that the mentioned objectives are not independent but interrelated with each other. For example, it is discovered that minimizing total delay can also minimize the total number of stops and, hence, is more suitable for implementation [47]. To consider multiple factors at the same time, special performance index (PI), which is usually defined as the weighted sum of several variables, can also be used in optimization model. For instance, Goodall et al. optimized signal timing plans by minimizing a combination of delay, stops, and decelerations [37]. There are also studies with the objective function considering the minimization of fuel consumption such as [48].

In the reviewed literatures, an unsolved problem is the selection of objective function under different application scenarios. The problem has been stated in a recent research [49].

3. Optimization Models for Traffic Signal Control under CV Environment

The optimization model, as the core part of a traffic signal control system, outputs the timing plan including cycle lengths and phase distribution and thus needs to be well designed to improve the control effect. In this section, we summarize the proposed optimization models into three types: rule-based models, mathematical programming-based models, and artificial intelligence-based models. Uncategorized models can be found in other models.

3.1. Rule-Based Models. Rule-based models are defined as the models which determine the control parameters through specific equations built based on certain optimization criteria. This type of models mostly inherited traditional fix-time control algorithms, replacing the historical traffic data with real-time traffic states awareness under the CV environment.

For isolated intersections, the classical approach to determine cycle lengths is based on the well-known Webster's equation [50], which is a function of lost times and critical flow ratios, for minimum delay cycle lengths:

$$C = \frac{1.5 \cdot L + 5}{1 - \left(1/X_C\right) \cdot \sum_{i=1}^{n} \left(v_i/s_i\right)}, \tag{1}$$

where C is the optimum cycle length; L is the sum of lost times for all phases in a cycle; n is the number of critical lane groups; $1/X_C$ is the desired degree of intersection utilization (1.0 for operation at full capacity). Then the green time is then distributed according to the lost time of all phases:

$$G = C - \sum_{i=1}^{n} \left(Y_i - L_i\right), \tag{2}$$

where G is the green time; Y_i is the yellow time per phase i; L_i is the lost time per phase i.

Webster's method is modified to adapt the CV environment in several studies. A typical example is the CATS system proposed by Maslekar et al. [51]. In this system, the cycle time is formulated by a modified Webster's equation:

$$C = \frac{1.5 \cdot L + 5}{1 - \sum \left(D/\mathrm{Ln}\right)}, \tag{3}$$

where D/Ln is the ratio of density D in the cluster to the length Ln of the cluster. A clustering algorithm is defined, which will assist in estimating density D of vehicles approaching an intersection. Moreover, two approaches, that is, C-DRIVE and MC-DRIVE, were adopted and compared for density estimation at intersections. The proposed system is compared with a classic pretimed control system and an adaptive control system by simulation. The simulations also show that the data convergence time and the communication delay between vehicles and traffic signals do not decrease the efficiency of the system. Similar studies include [22, 41, 52].

More extensions reflect in estimating the traffic states in the near future and developing network-wide control. Lee et al. [32] proposed a cumulative travel-time responsive (CTR) real-time intersection control algorithm. The core of the algorithm is based on a stochastic state estimation technique utilizing Kalman filtering that is used in estimating the cumulative travel times under imperfect market penetration rates at every update interval. Compared to an optimized actuated control algorithm, the proposed CTR algorithm improved the total delay time and average speed of the intersection by 34 and 36%, respectively, at 100% market penetration. Bani Younes et al. [53] designed an ITL scheduling algorithm (ITLC) which utilizes vehicular ad hoc communications technology to gather the real-time traffic characteristics of all competing flows of traffic at each signalized road intersection. Further, the ATL algorithm was introduced for open-network control scenarios aiming at high traffic fluency for the arterial flows. Evaluations revealed that the ATL algorithm decreases the average queuing delay at each traffic light by 10% compared with previously introduced traffic scheduling systems. Lin et al. [54] proposed an algorithm that limits the boundary flow of a road network based on MFD and controls the maximum queuing length of each boundary section to avoid the overflow phenomenon.

Rule-based models have the advantages of conciseness and computational convenience. However, the coarse-grained modeling usually contains too much simplification, which compromises the accuracy of the formulation.

3.2. Mathematical Programming-Based Models. Mathematical programming, which means the selection of a best element with regard to some criterion from the feasible region [55], is the most commonly used method to optimize the timing plan of intersection signal control. As

a classic paradigm in the field of transportation research, considerable studies have made great contributions in this regard. Some representative studies are reviewed below.

Zhu and Ukkusuri [56] developed a linear programming formulation for autonomous intersection control (LPAIC) accounting for traffic dynamics within a CV environment. Firstly, a lane-based bilevel optimization model is introduced to propagate traffic flows in the network, accounting for dynamic departure time, dynamic route choice, and autonomous intersection control in the context of system optimum network model. Then the bilevel optimization model is transformed to the linear programming formulation by relaxing the nonlinear constraints with a set of linear inequalities. Also, the LPAIC formulation propagates traffic flow consistently as LTM and CTM. The control system is tested in both isolated and grid network scenarios. The simulation result shows that the proposed LPAIC algorithm produces lower total travel time in different V/C than an actuated longest queue first signal control algorithm.

Feng et al. [57] presented a real-time adaptive signal phase allocation algorithm using connected-vehicle data. The proposed algorithm optimizes the phase sequence and duration by solving a two-level optimization problem to minimize total vehicle delay and queue length. A real-world intersection is modeled in VISSIM to validate the algorithms. The results show that the proposed control algorithm outperforms actuated control by reducing total delay by as much as 16.33% in a high penetration rate case.

Li et al. [48] develop a modeling framework for optimizing the timing of a set of traffic signals by considering individual vehicle characteristics such as fuel consumption and travel time. The proposed strategy applies the intelligent driving model (IDM) to predict vehicle trajectories under the connected-vehicle environment. The resulting model is a mixed-integer nonlinear program. Mixed-integer linear or nonlinear model is also frequently adopted in numerous studies, especially studies referring to priority control [39, 58–60].

Priemer and Friedrich [61] presented a traffic control strategy which is phase-based as traditional traffic signal control methods but operates without common parameters like cycle times, offsets, or other fixed timings. In each discrete interval of 5 s, the control algorithm forecasts the future queue length for the next 20 seconds by listening to the received vehicles position and speed data. Within each of these optimization horizons, the method determines the optimal phase sequence in order to reduce the total queue length at an intersection by using the methods of dynamic programming (DP) and complete enumeration (CE) algorithm. Approximate dynamic programming, a variant DP method which allows the controller to learn from its own performance progressively, is adopted in a study by Cai et al. [62]. This study presents a method, VICAC, which combines travel-time estimation and adaptive traffic signal control. Li and Ban [63] decomposed the signal optimization and coordination problem into two levels: an intersection level to optimize phase durations using dynamic programming (DP) and a corridor level to optimize the offsets of all intersections.

Pandit et al. [36] formulate the vehicular traffic signal control problem as a job scheduling problem on processors, with jobs corresponding to platoons of vehicles. Then the oldest job first (OJF) algorithm is used to minimize the delay across the intersection. The evaluation result shows that the algorithm reduces the delays experienced by vehicles as they pass through the intersection under light and medium traffic loads as compared with vehicle-actuated methods, Webster's method, and pretimed signal control methods.

For mathematical programming, the greatest challenge is the nonlinearities of objective functions and constraints. When the objective functions or constraints are complex, we may turn to heuristic algorithms or dynamic programming method, yet the computational burden is a potential problem for real-time control.

3.3. Artificial Intelligence-Based Models. Artificial intelligence (AI) is defined as the study of intelligent agents which perceives its environment and takes actions that maximize its chance of success at some goal. Traffic signal control based on AI is considered as a promising research area, because an AI system does not assume any prior knowledge of a model or the parameters of its dynamics and thus does not need to rely on the expensive and time-consuming model calibration procedures like existing optimization model. Furthermore, the AI system is capable of "learning" from the environment and thus is expected to improve the operations over time and adapt to changes of the environment. This has been proved by some leading researches in recent years [64, 65]. This type of models is very applicable to the traffic signal control under CV environment for the reason that the traffic information can be perceived in real time. Examples of AI approaches used in reviewed studies include multiagent system, fuzzy logic, neural network, and reinforcement learning.

Both vehicles and traffic signal controllers can be deemed as "agent," named as vehicle agent and control agent. In this perspective, the whole signal control system becomes a so-called multiagent system. The multiagent system used in a lot of studies has advantages including being model-free and coordinated, colearning, and being suitable for parallel processing. Khamis and Gomaa [66] developed an adaptive multiobjective reinforcement learning system for traffic signal control based on a cooperative multiagent framework. In addition, the authors show that using the Bayesian probability interpretation to estimate the parameters of the MDP probabilities can result in a good response to the traffic nonstationarity. Lee and Park [38] proposed a cooperative vehicle intersection control (CVIC) algorithm based on multiagent model that does not require a traffic signal. By eliminating the potential overlaps of vehicular trajectories coming from all conflicting approaches at the intersection, the CVIC algorithm seeks a safe maneuver for each vehicle approaching intersection and manipulates each of them. Also, an additional algorithm was designed to deal with the system failure cases resulting from inevitable trajectory overlaps at the intersection and infeasible solutions. A simulation-based case study implemented on a hypothetical

four-way single-lane approach intersection under varying congestion conditions showed that the CVIC algorithm significantly improved intersection performance compared with conventional actuated intersection control: 99% and 33% of stop delay and total travel time reductions, respectively, were achieved. Kari et al. [23] developed an agent-based online adaptive signal control (ASC) strategy. The system demonstrated savings of 5–14% in reducing travel time and 0–5% in reducing system-wide fuel economy in a scenario with constant demand profiles compared with the QEM/HCM-based strategy. Other representative studies include [67, 68].

Reinforcement learning (RL) is a machine learning paradigm, by which a controller's policy can be optimized through trial-and-error interactions with an environment. The agent-environment interaction can be described as follows. At every time step t, the agent obtains the state of environment s_t. Given s_t, the agent will decide the next action a_t. Then the environment transits to a new state, s_{t+1}, for which a reward r_{t+1} is given to the agent. The goal of reinforcement learning problem is to find the optimal policy which yields the highest total reward. A schematic illustration is shown in Figure 1. Reinforcement learning is particularly well suited to problems which include a long-term versus short-term reward trade-off. It has been applied successfully to various problems, including robot control, elevator scheduling, telecommunications, backgammon, checkers, and go (AlphaGo).

Liu et al. [69] presented distributed cooperative reinforcement learning-based traffic control that integrates V2X networks' dynamic clustering algorithm. A dynamic clustering algorithm is proposed based on the enhanced affinity propagation. By integrating the clustering algorithm, a cooperative reinforcement learning control scheme is proposed to balance the traffic load. To address the tough dimensionality curse of reinforcement learning, a distributed mechanism for intersection cooperation is introduced, and a fast gradient-descent function approximation method is proposed to improve the controls' real-time performance. The proposed algorithm gets the minimum intersection waiting time, serves the most road users, and produces the minimum queue length compared to the baselines. Cheng et al. [44] applied reinforcement learning to fine-tune the control parameters of the network and make it adaptive to various traffic conditions. The algorithm has good performance, especially in high dynamic traffic flows. Yang and Tan [70] provided some results on how the reinforcement learning method performs using Q-matrix and Q-network.

Wang et al. [71] formulated the joint traffic signal and connected vehicle control problem as a reinforcement learning (RL) problem, the action and state spaces of which are specifically designed to take into account the connected vehicles. An effective rewarding mechanism is designed, which takes into account the impact of the detouring on the network traffic efficiency. By utilizing tools from deep RL, an efficient algorithm is proposed to jointly control the traffic signals and the connected vehicles. Numerical results demonstrate validity and efficiency of the models.

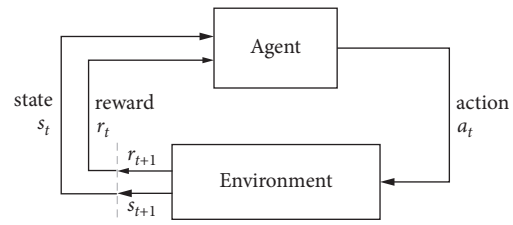


Figure 1: Schematic figure for reinforcement learning.

Another commonly used intelligent-control model is fuzzy logic—a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively). Collotta et al. [33] proposed a multicontroller system consisting of a Wireless Sensor Network (WSN), a Phase Sorting Module, and a fuzzy logic controller. The WSN is responsible for the collection of traffic data; the Phase Sorting Module determines the phase execution sequence; each fuzzy controller determines the green time duration for the relevant phase considering the number of enqueued cars in the lanes that are under its control. The fuzzy control strategy is also adopted in [44]. The authors determined the appropriate groups based on real-time traffic conditions using neurofuzzy network.

Before AI-based signal control system is widely implemented, it still faces significant hurdles, despite a large number of studies. A key problem is the lack of interpretability for AI algorithms, especially machine learning algorithms. In addition, some of AI algorithms require tremendous computing power.

3.4. Other Models. Ahmane et al. [45] proposed a model based on Timed Petri Nets with Multipliers (TPNM) which can make the control policy through the structural analysis. The control aims to smooth the traffic through the sequence of vehicles authorized to traverse the intersection. The proposed control policy is based on the modeling of an isolated 4-way intersection as a discrete event dynamic system. The dynamic behavior of the modeled traffic is discrete and represented by a Timed Petri Net with Multipliers (TPNM) in which multipliers are associated with the arcs of the Petri Net. Cheng et al. [72] used the game-theoretic paradigm of fictitious play to iteratively search for a coordinated signal timing plan to be employed, which improves a system-wide performance criterion for a traffic network. Elhenawy et al. [5] proposed a game-theory-based algorithm for controlling autonomous vehicle movements at uncontrolled intersections.

4. Major Technical Issues

4.1. Different Stages of CV Technologies. According to whether or not autonomous driving technology is equipped, connected vehicles can be divided into manned and automated types. The two types need to be controlled in different ways: manned connected vehicles might be still controlled

using traffic signals just as conventional vehicles that cannot communicate with the central controller or other vehicles, while autonomous vehicles can be controlled by coordinated route planning without any traffic signal. In most of the reviewed studies such as [22, 36, 57], the connected vehicles only refer to the manned connected vehicles. There are also studies that aim to solve the route planning problem in a fully autonomous vehicle environment [38, 44]. The situation of two types of vehicles together with conventional vehicles has not been fully studied. One example is the study of [73], in which three categories of vehicles are considered: (1) conventional vehicles, (2) automated vehicles, and (3) manned connected vehicles. The study integrated three different stages of technology development and developed heuristics to switch the signal controls depending on the stage of technology. The simulation results show an evident decrease in the total number of stops and delay when using the connected-vehicle algorithm for the tested scenarios with information level as low as 50%.

Rafter et al. [74] proposed a novel traffic signal control algorithm called Multimode Adaptive Traffic Signals (MATS) which can offer reductions in mean delay for networks with 0–100% connected-vehicle presence. The MATS algorithm combines position information from connected vehicles with data obtained from existing inductive loops and signal timing plans in the network to perform decentralized traffic signal control at urban intersections. The MATS algorithm is capable of adapting to scenarios with low numbers of connected vehicles, an area where existing traffic signal control strategies for connected environments are limited.

4.2. Centralized Control versus Decentralized Control. Majority of signal control systems use centralized formulation and architecture. In such systems, vehicles approaching the intersection communicate with a central controller at the intersection. The central controller optimizes various signal timing parameters of the system at the same time in one mathematical program. However, network signal timing optimization is known as an NP-complete problem and a central optimization technique will not be scalable and applicable to large transportation networks.

The other category of methods uses decentralized control (also called distributed control). Compared with centralized system, the paradigm decomposes the signal timing optimization problem to several interconnected subproblems in different control nodes. Each node only needs to execute a very simple computation, which reduces the complexity of each individual node. For example, Nafi et al. [75] presented a VANET-based road traffic signaling system developed using a distributed architecture by incorporating the distributed networking feature. Ishlam et al. [76] presented a Distributed-Coordinated methodology for signal timing optimization in connected urban street networks, with underlying assumption that all vehicles and intersections are connected, and intersections can share information with each other. The novelty of the work is the decentralized approach, where a mathematical program controls the timing of only a single intersection, which means the approach is in real time and scalable. The results show that the algorithm can increase intersection throughput between 1% and 5% and reduce travel time between 17% and 48%, compared to actuated coordinated signals. Decentralized architecture is also adopted in [5, 37, 51]. But the distributed system has also disadvantages: using distributed cooperative control completely could cause delays in realizing the traffic balance among districts. Therefore, a potential direction is introducing a hierarchical structure to incorporate the centralized control and the distributed cooperative control.

4.3. Isolated Control versus Coordinated Control. There are two distinct modes of traffic signal controller operation: isolated and coordinated. Many studies focus on isolated control because this mode is the foundation of intersection signal control and a significant part of coordinated mode. Liang et al. [77] developed a flexible, real-time traffic signal control algorithm to optimize both phase durations and phase sequences at four-approach intersections with conflicting left turns, based on information obtained from connected vehicles. The location of all connected vehicles is used to identify the presence of nonconnected vehicles that are stopped at the intersection and then identify naturally occurring platoons in the traffic stream. The signal control algorithm then selects the optimal sequence that these platoons should discharge through the intersection to minimize average delay of all identified vehicles. Several heuristic methods are proposed to determine optimal platoon departure sequences in this scenario. Other examples include [22, 37, 47].

The goal of coordinated control is to achieve the global optimum in an arterial or a network, through considering the coordination of all the controllers. Wang et al. [78] developed a joint control model which optimizes the speeds of the connected vehicles and coordinating signals along an arterial simultaneously. This control model forms connected vehicles into platoons so that the vehicles can pass through intersections together with no stops or the least stop time. At the same time, it optimizes signal timing plans along an arterial to achieve lower signal delay and higher throughput. A real-world road network simulation shows that the joint control model can reduce the stop time and stops of coordinate phase by up to 53.69% and 41.15%. The signalized intersection delay per vehicle is reduced by 13.19%. Other representative studies include [39, 68].

4.4. Analysis of Scenarios of CV Deployment with Different Penetration Rates. The quality of traffic signal control under CV environment depends mainly on the number of vehicles equipped with communication devices with respect to the total number of vehicles, the so-called penetration rate. Therefore, various penetration rates should be modeled to obtain the impact factor exactly. He et al. [39] evaluated the performance of the proposed PAMSCOD system and concluded that the system can outperform state-of-practice signal control methods at about a 40% penetration rate. Feng et al. [57] presented a real-time adaptive signal phase

allocation algorithm using connected-vehicle data. Due to the low penetration rate of the connected vehicles, an algorithm, EVLS, which estimates the states of unequipped vehicle based on connected-vehicle data is developed to construct a complete arrival table for the phase allocation algorithm. Evaluation shows that the proposed control algorithm outperforms actuated control by reducing total delay by as much as 16.33% in a high penetration rate case and similar delay in a low penetration rate case. Guler et al. [47] found that increase in the penetration rate from 0% up to 60% can significantly reduce the average delay for the proposed algorithm. The above-mentioned studies reveal that the existing control models can outperform the traditional control methods only in a relatively high penetration rate of connected vehicles, which may not be feasible in the near future. How to utilize real-world CV data under low penetration rate environment to improve traffic signal operation is a pressing issue. One of the related studies is that of Zheng and Liu [79]. The authors modeled vehicle arrivals at signalized intersections as a time-dependent Poisson process. An expectation maximization (EM) procedure is derived to solve the parameter estimation problem.

4.5. Priority Control. Priority control aims to improve service and reduce delay for certain traffic mode at intersections. The most common form is transit signal priority (TSP) control. Hu et al. [59] proposed a person-delay-based optimization method for an intelligent TSP logic TSPCV-C that enables bus/signal cooperation and coordination among consecutive signals under the CV environment. The method is evaluated through a computer simulation as well as in the field [60]. A similar form is freight signal priority (FSP). Rather than considering simply travel time and reliability, it is possible to consider vehicle weight, road grade, and truck engine type in order to minimize energy and emissions along a freight corridor [23].

The concept can be generalized for a system in which different traffic modes have different priority level. He et al. [39] presented a unified platoon-based formulation called PAMSCOD to concurrently optimize network traffic signal control for different travel modes given the assumption that advanced communication systems are available between vehicles and traffic controllers. Two modes of traffic composition (transit buses and passenger vehicles) are considered in a decision framework. Microscopic simulation shows that the proposed algorithm can successfully coordinate traffic signals considering the two traffic modes including buses and automobiles and significantly reduce vehicle delay for both modes. The algorithm was improved by the authors in another paper [58]. Liang et al. [80] proposed an algorithm that leverages information from connected vehicles (CVs) arriving at an intersection to identify naturally occurring platoons that consist of both CVs and non-CVs. Simulation tests reveal that the proposed platoon-based algorithm provides superior computational savings (over 95%) compared with algorithms that focus on individual vehicles.

4.6. Performance Evaluation. There are four major elements in the performance evaluation for traffic control optimization: (1) measure of effectiveness (MOE), (2) evaluation platform, (3) evaluation scenarios, and (4) baselines. The widely used MOEs include average delay, stops, queue length, and energy consumption. Most studies use simulation software as evaluation platform such as VISSIM, SUMO, GLD, AIMSUN NG, and Commuter for microscopic traffic simulation and ns2, ns3, NCTUns, and OPNET for network communication simulation. Simulation experiments with different scenarios are carried out by covering varying volume-to-capacity ratios and market penetration rates in most studies. The baselines are usually classic algorithms such as Webster's method, TRANSYT, HCM method, and Synchro's optimization method.

It is noted that the car-following model in a connected-vehicle environment may be different from traditional models so appropriate adjustments of the default settings in traffic simulation software are necessary. However, no details about the adjustments are mentioned in the existing studies.

5. Summary

To demonstrate the development status of the research area more clearly, we summarize the basic information of the reviewed papers in Table 1.

6. Perspectives of Future Development

Connected vehicle is a promising technology which provides more detailed traffic information to optimize time planning of signalized intersections in real time. It has been proved that the introduction of CV technology into traffic signal control has potential in improving the road traffic flow efficiency, providing enhanced safety, saving energy consumption, and reducing pollutant emissions. The paper reviews the existing studies about time signal optimization under CV environment to illustrate the current state of the art in this research field. A variety of optimization models have been proposed and evaluated, achieving satisfying performance in either simulation or implementation. However, it is clear that the application of CV technology in the traffic signal optimization domain is still in its early stages. There remain many problems that are waiting to be solved. In this section, we raise unresolved problems and give our perspectives of the future research directions.

Under CV environment, the traffic flow characteristics can be quite different from those in current road traffic environment. Furthermore, things may become more complex due to the coexistence of conventional vehicles and manned connected vehicles together with connected autonomous vehicles. Therefore, the flow pattern under CV environment needs to be taken into account. In addition, signal control optimization can be incorporated with the help of driver assistant system in such environment to further reduce delays.

The current studies adopt either centralized or decentralized (or distributed) paradigm to optimize signal time

Table 1: Summary of the reviewed papers.

Year	Authors	Title	Method	Optimization objectives	Isolated/ coordinated	Centralized/ decentralized	Penetration rate	Evaluation platform
[38]	Lee et al.	Development and evaluation of a cooperative vehicle intersection control algorithm under the connected-vehicles environment	CVIC algorithm	Minimize total length of overlapped trajectories	Isolated	Centralized	100%	VISSIM
[36]	pandit et al.	Adaptive Traffic Signal Control with Vehicular Ad hoc Networks	Job scheduling	Minimize the delay across the intersection	Isolated	Centralized	100%, 90%, 70%, 50%, 30%	SUMO and OMNET++/INET
[22]	Gradinescu et al.	Adaptive Traffic Lights Using Car-to-Car Communication	Webster's formula	Minimize the delay at intersections	Isolated	Centralized	100%	VISSIM and ns2, Jist/SWANS
[39]	He et al.	PAMSCOD : Platoon-based Arterial Multi-modal Signal Control with Online Data	PAMSCOD formulation	Minimize the weighted delay as well as the total green rest time	Arterial	Centralized	100%, 80%, 60%, 40%, 20%	VISSIM
[58]	He et al.	Multi-modal traffic signal control with priority, signal actuation and coordination	A request-based MILP formulation	Minimize the weighted delay for active priority requests, passenger vehicles, and virtual coordination requests	Arterial	Centralized	Only priority eligible traffic modes are equipped with V2I communication system (passenger cars can only actuate the signals via loop detectors)	VISSIM
[48]	Li et al.	Traffic signal timing optimization in connected vehicles environment	A green times optimization algorithm	Minimize the weighted sum of total system fuel consumption and travel time	Isolated	Centralized	100%	VISSIM
[37]	Goodall et al.	Traffic Signal Control with Connected Vehicles	PMSA algorithm	Minimize either delay only or a combination of delay, stops, and decelerations	Isolated	Decentralized	100%, 50%, 25%, 10%	VISSIM
[60]	Lee et al.	Transit Sgnal Priority Experiment in a Connected-Vehicle Technology Environment	TSPCV algorithm	Minimize per person delay	Arterial	Centralized	Only buses are equipped with V2I communication system	—
[59]	Hu et al.	Coordinated transit signal priority supporting transit progression under Connected-Vehicle Technology	TSPCV algorithm	Minimize per person delay	Arterial	Centralized	Only buses are equipped with V2I communication system	VISSIM
[5]	Elhenawy et al.	An Intersection game-theory-Based Traffic Control Algorithm in a Connected Vehicle Environment	A game-theory-based algorithm	—	Isolated	Decentralized	100%	—

Table 1: Continued.

Year	Authors	Title	Method	Optimization objectives	Isolated/ coordinated	Centralized/ decentralized	Penetration rate	Evaluation platform
[73]	Yang et al.	Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles	A bilevel mathematical program algorithm	Minimize total delay	Isolated	Centralized	Multiple scenarios	—
[70]	Yang et al.	A reinforcement learning-based traffic signal control algorithm in a connected vehicle environment	Q-learning algorithm	Minimize total delay	Isolated	Centralized	100%	SUMO
[51]	Maslekar et al.	CATS: An adaptive traffic signal system based on car-to-car communication	C-DRIVE and MC-DRIVE	Reduce average waiting time and queue length	Isolated	Decentralized	100%	NCTUns
[62]	Cai et al.	Vehicle-to-infrastructure communication-based adaptive traffic signal control	VICAC algorithm	Minimize travel time	Isolated	Centralized	100%	Commuter
[63]	Li et al.	Connected vehicles based traffic signal timing optimization	A CV-based traffic signal optimization framework	Minimizing fuel consumption and travel time	Arterial	Decentralized	100%	SYNCHRO
[74]	Rafter et al.	Augmenting traffic signal control systems for urban road networks with connected vehicles	Multimode Adaptive Traffic Signals (MATS)	Optimize network capacity for saturated conditions and minimize stops and delays for undersaturated conditions	Arterial	Decentralized	0–100%	SUMO
[75]	Nafi et al.	A VANET Based Intelligent Road Traffic Signaling System	IRTSS system	Reduce travel time and gas emissions	Isolated	Decentralized	100%	OPNET Modeler
[76]	Islam et al.	Distributed-coordinated signal timing optimization in connected transportation networks	A Distributed-Coordinated methodology	Maximize the intersection throughput while penalizing for queue length	Network	Decentralized	100%	VISSIM
[57]	Feng et al.	A real-time adaptive signal control in a connected vehicle environment	EVLS algorithm	Minimize total vehicle delay or minimize the queue length	Isolated	Centralized	100%, 75%, 50%, 25%	VISSIM
[48]	Chandan et al.	Real-time Traffic Signal Control for Isolated Intersection, using Car-following Logic under Connected-Vehicle Environment	A queue-length based optimization algorithm	Minimize the travel time delay and average number of stops per vehicle	Isolated	Centralized	100%	VISSIM
[47]	Guler et al.	Using connected vehicle technology to improve the efficiency of intersections	Platooning based algorithm	Minimize total delay and minimize number of stops	Isolated	Centralized	100%, 80%, 60%, 40%, 20%, 0%	MATLAB

Table 1: Continued.

Year	Authors	Title	Method	Optimization objectives	Isolated/ coordinated	Centralized/ decentralized	Penetration rate	Evaluation platform
[44]	Cheng, et al.	Fuzzy Group-Based Intersection Control via Vehicular Networks for Smart Transportations	A fuzzy group-based control algorithm	Reduce AWT (average waiting time) and WTV (waiting time variance)	Isolated	Decentralized	100%	ns3
[36]	Pandit et al.	A study on traffic signal control at signalized intersections in vehicular ad hoc networks	A queue-length based optimization algorithm	Minimize delay and queue length	Isolated	Centralized	100%	GLD
[32]	Lee et al.	Cumulative Travel-Time Responsive Real-Time Intersection Control Algorithm in the Connected-Vehicle Environment	CTR algorithm	Decrease total travel time, improve average speed, and maximize throughputs	Isolated	Centralized	100%, 90%, 70%, 50%, 30%, 10%	VISSIM
[79]	Zheng et al.	Estimating traffic volumes for signalized intersections using connected vehicle data	A probability theory-based algorithm	—	—	—	Very low	—
[48]	Li et al.	Traffic Signal Timing Optimization Incorporating Individual Vehicle Fuel Consumption Characteristics under Connected Vehicles Environment	A mixed-integer nonlinear programming model	Minimize the total system travel and fuel consumption	Isolated	Centralized	100%	VISSIM
[61]	Priemer et al.	A Decentralized Adaptive Traffic Signal Control Using V2I Communication Data	A dynamic programming model	Minimize the total queue length	Isolated	Decentralized	100%, 50%, 33%, 25%, 20%, 17%, 14%, 12%, 10%	AIMSUN NG
[69]	Liu et al.	Distributed Cooperative Reinforcement Learning-Based Traffic Signal Control That Integrates V2X Networks' Dynamic Clustering	Reinforcement learning	Reduce total intersection waiting time and queue length	Isolated	Decentralized	100%	ns3 and SUMO
[54]	Lin et al.	Improved Road-Network-Flow Control Strategy Based on Macroscopic Fundamental Diagrams and Queuing Length in Connected-Vehicle Netwosrk	A MFD based algorithm	Avoid oversaturating in the road network	Network	Centralized	100%	VISSIM
[77]	Liang et al.	A heuristic method to optimize generic signal phasing and timing plans at signalized intersections using Connected-Vehicle technology	A real-time traffic signal control algorithm	Optimize both phase durations and phase sequences	Isolated	Centralized	Less than 100%	—

Table 1: Continued.

Year	Authors	Title	Method	Optimization objectives	Isolated/ coordinated	Centralized/ decentralized	Penetration rate	Evaluation platform
[78]	Wang et al.	A joint control model for connected vehicle platoon and arterial signal coordination	A joint control model	Minimize vehicle stop time	Arterial	—	—	VISSIM/ MATLAB
[68]	Xiang et al.	An adaptive traffic signal coordination optimization method based on vehicle-to-infrastructure communication	A multiagent-based model	Minimize delay, travel time, and queue length	Network	Centralized	100%	VISSIM
[23]	Kari et al.	Development of an Agent-Based Online Adaptive Signal Control Strategy Using Connected-Vehicle Technology	ATSC algorithm	Reduce delay and fuel consumption	Isolated	Centralized	100%	SUMO
[56]	Zhu et al.	A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment	LPAIC algorithm	Minimize total travel time	Isolated	Centralized	100%	Unknown
[45]	Ahmane et al.	Modeling and controlling an isolated urban intersection based on cooperative vehicles	A TPNM based algorithm	Minimize queue length	Isolated	Centralized	100%	
[46]	Shaghaghi et al.	Adaptive green traffic signal controlling using vehicular communication	TSCS algorithm	Decrease the vehicle waiting time and pollutant emissions at intersections	Isolated	Centralized	100%	OMNET++ and SUMO
[53]	Bani Youne et al.	Intelligent Traffic Light Controlling Algorithms Using Vehicular Networks	ITLC algorithm	Reduce the expected queuing delay time of each vehicle and increase the throughput of the intersection	Network	Centralized	100%	ns2
[40]	Tomescu et al.	Adaptive traffic light control system using ad hoc vehicular communications network	A rule-based model	Reduce number of stops and delay	Arterial	Centralized	100%	MATLAB
[71]	Wang et al.	Joint Traffic Signal and Connected-Vehicle Control in IoV via Deep Reinforcement Learning	Deep reinforcement learning	Detouring ratio and network traffic efficiency	Network	Centralized	—	SUMO

Table 1: Continued.

Year	Authors	Title	Method	Optimization objectives	Isolated/ coordinated	Centralized/ decentralized	Penetration rate	Evaluation platform
[33]	Collotta et al.	A novel approach for dynamic traffic lights management based on Wireless Sensor Networks and multiple fuzzy logic controllers	A fuzzy logic-based model	Reduce average waiting time and queue length	Isolated	Centralized	100%	Simulink/ MATLAB
[66]	Khamis et al.	Adaptive multiobjective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multiagent framework	A multiagent-based model	Minimize average trip waiting time (ATWT), average trip time (ATT), and average junction waiting time (AJWT) and maximize flow rate (FR)	Network	Centralized	100%	GLD
[72]	Cheng et al.	CoSIGN : A Parallel Algorithm for Coordinated Traffic Signal Control	A game-theory-based algorithm	Minimize the total travel time	Network	Centralized	100%	INTEGRATION-UM
[34]	Abbas et al.	High accuracy traffic light controller for increasing the given green time utilization	DT3P algorithm	Reducing the average and maximum queue lengths	Isolated	Centralized	100%	SIDRA
[42]	Kwatirayo et al.	Optimizing Intersection Traffic Flow Using VANET	ATLC algorithm	Reduce average waiting time	Isolated	Centralized	100%	SUMO
2009	Gurcan Comert et al.	Queue length estimation from probe vehicle location and the impacts of sample size	A probability theory-based algorithm	—	—	—	0, 5%, 10%, . . ., 100%	—
[52]	Chen et al.	A Realtime Dynamic Traffic Control System Based on Wireless Sensor Network	A rule-based model	Minimize average waiting time	Isolated	Centralized	100%	—
[41]	Bhuvaneswari et al.	Adaptive Traffic Signal Flow Control using Wireless Sensor Networks	ATSWSN algorithm	Reduce the waiting time	Isolated	Centralized	100%	LabView
[67]	Kari et al.	Eco-Friendly Freight Signal Priority Using Connected-Vehicle Technology: A Multi-Agent Systems Approach	A multiagent-based model	Reducing energy and emissions	Isolated	Centralized	Only priority vehicles are equipped with V2I communication system	SUMO
[80]	Liang et al.	Signal timing optimization with connected vehicle technology: Platooning to improve computational efficiency	A real-time traffic signal optimization algorithm	Minimize total vehicle delay	Isolated	—	0, 20%, 40%, . . ., 100%	—

planning. Centralized control can achieve the coordination of different intersections easily, leading to a global optimal solution, yet the formulation might be computation-expensive in a large road network. In contrast, decentralized control is more scalable but inefficient in realizing coordination. The future traffic signal control system should incorporate the advantages of the two modes to achieve general applicability and higher efficiency.

Under the connected-vehicle environment, the greatest strength is the complete perception of the traffic states in the road network. The cooperation of different intersections, therefore, should be improved. But the study of network-wide signal optimization is insufficient. We take network control under CV environment as an important research direction.

Despite the rapid popularity of CV technology, the situation of imperfect penetration rate may last for a long time. Thus, the state estimation of unequipped vehicles (especially in a low penetration rate) will be a crucial problem at present. Although Zheng et al. [79] proposed a probability theory-based model to deal with the problem, the sphere of application of the method is still limited.

As shown in several studies, for example, [37], the algorithm can improve performance compared to a state-of-practice coordinated actuated timing plan at low- and mid-level volumes, but performance worsens at oversaturated condition. The optimization strategies for oversaturated traffic are expected to be solved under CV environment.

In recent years, the field of artificial intelligence has achieved great breakthrough with the advance of deep learning and reinforcement learning. It is not surprising that the artificial intelligence algorithm has been introduced into traffic signal control and used for solving complex control optimization problems to which mathematical or conventional modeling is unlikely to be useful. We consider the artificial intelligence-based control as the future trend of traffic signal optimization.

Data Availability

All the reviewed papers in the manuscript are listed in Table 1.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study was jointly supported by Shanghai Science and Technology Committee, China (YDZX20193100004845, 21DZ205100, and 19DZ1208700).

References

[1] D. Schrank, B. Eisel, T. Lomax, and J. Bak, "Urban mobility report," *Inrix*, vol. 81, 2015.

[2] NHTSA, "Traffic safety facts 2015: a compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system," *U.S. Department of Transportation - NHTSA - DOT HS 809 322*, 2015, http://www.trb.org/main/blurbs/172686.aspx.

[3] M. Desai and R. P. Harvey, "Inventory of U.S. Greenhouse gas emissions and sinks: 1990-2015. Fed," *Register*, vol. 82, Article ID 10767, 2017.

[4] A. Pizam, "Life and tourism in the year 2050," *International Journal of Hospitality Management*, vol. 18, pp. 331–343, 1999.

[5] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 343–347, Gran Canaria, Spain, September 2015.

[6] F. V. Webster and B. M. Cobbe, "Traffic signals," vol. 56, H. M. S. O., London, UK, 1966, Road Research Technique Paper.

[7] R. E. Allsop, "SIGSET: a computer program for calculating traffic signal settings," *Traffic Engineering and Control*, vol. 13, pp. 58–60, 1971.

[8] R. E. Allsop, "SIGCAP: a computer program for assessing the traffic capacity of signal-controlled road junctions," *Traffic Engineering and Control*, vol. 17, 1976.

[9] J. D. C. Little, M. D. Kelson, and N. H. Gartner, "MAXBAND: a program for setting signals on arteries and triangular networks," *Transportation Research Record Journal of the Transportation Research Board*, vol. 795, pp. 40–46, 1981.

[10] D. I. Robertson, "TRANSYT method for area traffic control," *Traffic Engineering and Control*, vol. 10, pp. 181-182, 1969.

[11] N. H. Gartner and R. M. Deshpande, "Dynamic programming approach for arterial signal optimization," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2356, no. 1, pp. 84–91, 2013.

[12] R. A. Vincent and J. R. Peirce, ""MOVA" - experience and developments in signal control," in *Proceedings of the PTRC TRAFFEX '93 Conference Proceedings," Seminar On Traffic Control: Control Systems, Information And Enforcement*, National Exhibition Centre, Birmingham, UK, April 1993.

[13] P. Koonce, L. Rodegerdts, K. Lee et al., *Traffic Signal Timing Manual*, 2008.

[14] A. J. Miller, "A computer control system for traffic networks," in *Proceedings of the International Symposium on the Theory of Traffic Flow and Transportation*, London, UK, 1963.

[15] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and R. I. Winton, *SCOOT-A Traffic Responsive Method of Coordinating Signals*, Transport and Road Research Laboratory, London, UK, 1981.

[16] A. G. Sims, "The Sydney coordinated adaptive traffic system," in *Proceedings of the Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems*, Pacific Grove, CA, USA, February 1979.

[17] J. J. Henry, J. L. Farges, and J. Tuffal, "The PRODYN real time traffic algorithm," *Control in Transportation Systems*, pp. 305–310, 1984.

[18] N. H. Gartner, "OPAC: a demand-responsive strategy for traffic signal control," *IFAC Proceedings Volumes*, vol. 93, pp. 241–244, 1990.

[19] V. Mauro and D. Di Taranto, "UTOPIA: proceedings of the 6th IFAC/IFIP," in *Proceedings of the IFORS Symposium on Control and Communication in Transportation*, Paris, France, September 1990.

[20] P. Mirchandani and F.-Y. Wang, "RHODES to intelligent transportation systems," *IEEE Intelligent Systems*, vol. 20, pp. 10–15, 2005.

[21] M. M. Zanjireh and H. Larijani, "A survey on centralised and distributed clustering routing algorithms for WSNs," in

Proceedings of the Vehicular Technology Conference, pp. 1–6, Boston, MA, USA, September 2015.

[22] V. Gradinescu, C. Gorgorin, R. Diaconescu, V. Cristea, and L. Iftode, "Adaptive traffic lights using car-to-car communication," in *Proceedings of the 2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pp. 21–25, Dublin, Ireland, April 2007.

[23] D. Kari, G. Wu, and M. J. Barth, "Development of an agent-based online adaptive signal control strategy using connected vehicle technology," in *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Shandong, China, October 2014.

[24] NHTSA, *Preliminary Statement of Policy Concerning Automated Vehicles*, National Highway Traffic Safety Administration, Washington, DC, USA, 2013.

[25] Y. Xie, H. Zhang, N. H. Gartner, and T. Arsava, "Collaborative merging strategy for freeway ramp operations in a connected and autonomous vehicles environment," *Journal of Intelligent Transportation Systems*, vol. 21, no. 2, pp. 136–147, 2017.

[26] Y. Wang, W. E, W. Tang, D. Tian, G. Lu, and G. Yu, "Automated on-ramp merging control algorithm based on Internet-connected vehicles," *IET Intelligent Transport Systems*, vol. 7, no. 4, pp. 371–379, 2013.

[27] R. A. Uzcátegui, A. J. De Sucre, and G. Acosta-Marum, "Wave: a tutorial," *IEEE Communications Magazine*, vol. 47, 2009.

[28] D. Llusia, R. Márquez, J. F. Beltrán, C. Moreira, and J. P. D. Amaral, "IEEE 802.11p: towards an international standard for wireless access in vehicular environments," in *Proceedings of the Vehicular Technology Conference*, pp. 2036–2040, Calgary, Canada, September 2008.

[29] D. Committee, "Dedicated short range communications (DSRC) message set dictionary," Technical Report J2735_200911, Society of Automotive Engineers, Warrendale, PA, USA, 2009.

[30] R. Braun, F. Busch, C. Kemper et al., "TRAVOLUTION--Netzweite Optimierung der Lichtsignalsteuerung und LSA-Fahrzeug-Kommunikation," *Stra{ß}enverkehrstechnik*, vol. 53, pp. 365–374, 2009.

[31] BMW, *BMW GREEN Wave Project*, [WWW Document]. URL http://www.themotorreport.com.au/46137/bmw-pushing-for-smarter-adaptive-traffic-lights, 2009.

[32] J. Lee, B. Park, and I. Yun, "Cumulative travel-time responsive (CTR) real-time intersection control algorithm under the connected vehicle environment," *Journal of Transportation Engineering*, vol. 139, Article ID 130605045950002, 2013.

[33] M. Collotta, L. Lo Bello, and G. Pau, "A novel approach for dynamic traffic lights management based on Wireless Sensor Networks and multiple fuzzy logic controllers," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5403–5415, 2015.

[34] M. K. Abbas, M. N. Karsiti, M. Napiah, B. B. Samir, M. Al-Jemeli, and H. Dag, "High accuracy traffic light controller for increasing the given green time utilization," *Computers & Electrical Engineering*, vol. 41, pp. 40–51, 2015.

[35] S. M. A. B. A. Islam, A. Hajbabaie, H. M. A. Aziz, and H. M. A. Aziz, "A real-time network-level traffic signal control methodology with partial connected vehicle information," *Transportation Research Part C: Emerging Technologies*, vol. 121, Article ID 102830, 2020.

[36] K. Pandit, C. Chuah, H. M. Michael, and D. Ghosal, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 1459–1471, 2013.

[37] N. J. Goodall, B. L. Smith, and B. Park, "Traffic signal control with connected vehicles," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2381, no. 1, pp. 65–72, 2013.

[38] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.

[39] Q. He, K. L. Head, and J. Ding, "PAMSCOD: platoon-based arterial multi-modal signal control with online data," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 164–184, 2012.

[40] O. Tomescu, I. M. Moise, and A. E. Stanciu, "Adaptive traffic light control system using ad hoc vehicular communications network," *UPB Scientific Bulletin, Series D*, vol. 74, no. 2, 2012.

[41] P. T. V. Bhuvaneswari, G. V. A. Raj, R. Balaji, and S. Kanagasabai, "Adaptive traffic signal flow control using wireless sensor networks," in *Proceedings of the 2012 Fourth International Conference on Computational Intelligence and Communication Networks*, pp. 85–89, Uttar Pradesh, India, November 2012.

[42] S. Kwatirayo, J. Almhana, and Z. Liu, "Optimizing intersection traffic flow using VANET," in *Proceedings of the 2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, pp. 260–262, New Orleans, LA, USA, June 2013.

[43] Z. Yao, Y. Jiang, B. Zhao, X. Luo, and B. Peng, "A dynamic optimization method for adaptive signal control in a connected vehicle environment," *Journal of Intelligent Transportation Systems*, vol. 24, no. 2, pp. 184–200, 2020.

[44] J. Cheng, W. Wu, J. Cao, and K. Li, "Fuzzy group-based intersection control via vehicular networks for smart transportations," *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 751–758, 2017.

[45] M. Ahmane, A. Abbas-Turki, F. Perronnet et al., "Modeling and controlling an isolated urban intersection based on cooperative vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 44–62, 2013.

[46] E. Shaghaghi, M. R. Jabbarpour, R. Md Noor, H. Yeo, and J. J. Jung, "Adaptive green traffic signal controlling using vehicular communication," *Frontiers of Information Technology and Electronic Engineering*, vol. 18, no. 3, pp. 373–393, 2017.

[47] S. I. Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121–131, 2014.

[48] K. Chandan, W. Li, X. J. Ban, and J. Wang, "Traffic signal timing optimization incorporating individual vehicle fuel consumption characteristics under connected vehicles environment," in *Proceedings of the 2016 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 13–18, Seattle, WA, USA, September 2016.

[49] A. Hajbabaie and R. F. Benekohal, "Traffic signal timing optimization: choosing the objective function," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2355, no. 19, 10 pages, 2013.

[50] F. V. Webster, "Traffic signal settings. road research," Department of Scientific and Industrial Research, London, UK, Technical Paper NO. 39, 1958.

[51] N. Maslekar, J. Mouzna, M. Boussedjra, and H. Labiod, "CATS: an adaptive traffic signal system based on car-to-car communication," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1308–1315, 2013.

[52] W. Chen, L. Lifeng Chen, Z. Zhanglong Chen, and S. Shiliang Tu, "A realtime dynamic traffic control system based on wireless sensor network," in *Proceedings of the 2005 International Conference on Parallel Processing Workshops (ICPPW'05)*, vol. 1, pp. 258–264, Oslo, Norway, June 2005.

[53] M. Bani Younes and A. Boukerche, "Intelligent traffic light controlling algorithms using vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 5887–5899, 2016.

[54] X. Lin, J. Xu, P. Lin, C. Cao, and J. Liu, "Improved road-network-flow control strategy based on macroscopic fundamental diagrams and queuing length in connected-vehicle network," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8784067, 7 pages, 2017.

[55] H. P. Williams, *Model Building in Mathematical Programming*, John Wiley and Sons, New York, NY, USA, 2013.

[56] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.

[57] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A real-time adaptive signal control in a connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 460–473, 2015.

[58] Q. He, K. L. Head, and J. Ding, "Multi-modal traffic signal control with priority, signal actuation and coordination," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 65–82, 2014.

[59] J. Hu, B. B. Park, and Y.-J. Lee, "Coordinated transit signal priority supporting transit progression under connected vehicle technology," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 393–408, 2015.

[60] Y.-J. Lee, S. Dadvar, J. Hu, and B. B. Park, "Transit signal priority experiment in a connected vehicle technology environment," *Journal of Transportation Engineering, Part A: Systems*, vol. 143, no. 8, Article ID 05017005, 2017.

[61] C. Priemer and B. Friedrich, "A decentralized adaptive traffic signal control using V2I communication data," in *Proceedings of the 2009 12th International IEEE Conference on Intelligent Transportation Systems*, pp. 765–770, St. Louis, MO, USA, October 2009.

[62] C. Cai, Y. Wang, and G. Geers, "Vehicle-to-infrastructure communication-based adaptive traffic signal control," *IET Intelligent Transport Systems*, vol. 7, no. 3, pp. 351–360, 2013.

[63] W. Li and X. Ban, "Connected vehicles based traffic signal timing optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4354–4366, 2018.

[64] E. V. D. Pol, "Deep reinforcement learning for coordination in traffic light control," Master Thesis, Faculteit der Natuurwetenschappen, Wiskunde en Informatica, Amsterdam, Netherlands, 2016.

[65] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[66] M. A. Khamis and W. Gomaa, "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 134–151, 2014.

[67] D. Kari, G. Wu, and M. J. Barth, "Eco - friendly freight signal priority using connected vehicle technology: a multi - agent systems approach," in *Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1187–1192, Dearborn, MI, USA, June 2014.

[68] J. Xiang and Z. Chen, "An adaptive traffic signal coordination optimization method based on vehicle-to-infrastructure communication," *Cluster Computing*, vol. 19, no. 3, pp. 1503–1514, 2016.

[69] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8667–8681, 2017.

[70] K. Yang and I. Tan, "A reinforcement learning based traffic signal control algorithm in a connected vehicle environment," in *Proceedings of the 17th Swiss Transport Research Conference*, Ascona, Switzerland, May 2017.

[71] Z. Wang, H. Zhu, Y. Zhou, and X. Luo, "Joint traffic signal and connected vehicle control in IoV via deep reinforcement learning," in *Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, Najing, China, April 2021.

[72] S.-F. Cheng, M. A. Epelman, and R. L. Smith, "CoSIGN: a parallel algorithm for coordinated traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 551–564, 2006.

[73] K. Yang, S. I. Guler, and M. Menendez, "Isolated intersection control for various levels of vehicle technology: conventional, connected, and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, 2016.

[74] C. B. Rafter, B. Anvari, S. Box, and T. Cherrett, "Augmenting traffic signal control systems for urban road networks with connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1728–1740, 2020.

[75] N. S. Nafi and J. Y. Khan, "A VANET based intelligent road traffic signalling system," in *Proceedings of the Australasian Telecommunication Networks and Applications Conference (ATNAC) 2012*, Brisbane, Australia, September 2012.

[76] S. M. A. B. A. Islam and A. Hajbabaie, "Distributed coordinated signal timing optimization in connected transportation networks," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 272–285, 2017.

[77] X. Liang, S. I. Guler, and V. V. Gayah, "A heuristic method to optimize generic signal phasing and timing plans at signalized intersections using connected vehicle technology," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 156–170, 2020.

[78] P. Wang, Y. Jiang, L. Xiao, Y. Zhao, and Y. Li, "A joint control model for connected vehicle platoon and arterial signal coordination," *Journal of Intelligent Transportation Systems*, vol. 24, no. 1, pp. 81–92, 2020.

[79] J. Zheng and H. X. Liu, "Estimating traffic volumes for signalized intersections using connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 347–362, 2017.

[80] X. Liang, S. I. Guler, and V. V. Gayah, "Signal timing optimization with connected vehicle technology: platooning to improve computational efficiency," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 18, pp. 81–92, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5439935, 10 pages
https://doi.org/10.1155/2021/5439935

Research Article

The Fusion of Multi-Focus Images Based on the Complex Shearlet Features-Motivated Generative Adversarial Network

Lei Wang,[1] **ZhouQi Liu**,[1] **Jin Huang**,[1] **Cong Liu**,[1] **LongBo Zhang**,[1] **and ChunXiang Liu**[2]

[1]*School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China*
[2]*Anhui Key Laboratory of Plant Resources and Plant Biology, Huaibei Normal University, Huaibei 235000, China*

Correspondence should be addressed to Lei Wang; wanglei0511@sdut.edu.cn

Received 29 June 2021; Accepted 15 July 2021; Published 30 July 2021

Academic Editor: Chi-Hua Chen

The traditional methods for multi-focus image fusion, such as the typical multi-scale geometric analysis theory-based methods, are usually restricted by sparse representation ability and the transferring efficiency of the fusion rules for the captured features. Aiming to integrate the partially focused images into the fully focused image with high quality, the complex shearlet features-motivated generative adversarial network is constructed for multi-focus image fusion in this paper. Different from the popularly used wavelet, contourlet, and shearlet, the complex shearlet provides more flexible multiple scales, anisotropy, and directional sub-bands with the approximate shift invariance. Therefore, the features in complex shearlet domain are more effective. With of help of the generative adversarial network, the whole procedure of multi-focus fusion is modeled to be the process of adversarial learning. Finally, several experiments are implemented and the results prove that the proposed method outperforms the popularly used fusion algorithms in terms of four typical objective metrics and the comparison of visual appearance.

1. Introduction

The target information may display the differentiation for the lengths of the focus during the imaging procedure, that is, the closer the object to the focus is, the clearer the image is. On the other hand, it is difficult to synchronously get the full-focus image by only one imaging device [1]. A common method to deal with this problem is to fuse multiple images of the same scene into images of different focal lengths, which is called the multi-focus image fusion and has been widely used in military monitoring, image analysis, and transportation [2]. For example, in modern wars, the multi-focus images can be used to monitor important targets and facilities of the enemy, and in the transportation domain, the multi-focus images can be used to track logistics and vehicle information and even penalize violations.

Nowadays, there are mainly four kinds of strategies for the fusion of multi-focus images: the spatial domain methods, the early transform domain methods, the multi-scale geometric analysis theory-based methods, and the deep learning theory-based methods. The spatial domain methods usually directly implement the linear computation on the image pixel, for example, the averaging method, maxing method, and weighted method. The early transform domain methods include the Laplace pyramid-based method, wavelet-based method, etc.

In these methods, the multi-focus images are decomposed into different scales and each scale is with a limited number of sub-bands. Then, the features in different levels can be obtained for fusion. For example, in reference [3], the authors proposed a fusion method by using the extremum of the wavelet coefficients in different sub-bands. Dou et al. [4] proposed a fusion method by using the region energy in different high-pass sub-band coefficients by considering their distributions. Due to the limited number of high sub-bands, some block artifacts of edges may appear in these methods. To deal with these problems, multi-scale geometric analysis theory-based methods have been popularly reported in recent years. The curvlet transform, contourlet transform, non-subsampled contourlet transform (NSCT), and the

shearlet are the typical decomposition tools in this period. For example, Li and Yang [5] proposed a fusion method by combining the wavelet and curvelet to overcome the disadvantages of the wavelet. He et al. [6] proposed a multi-focus image fusion method based on the improved contourlet package. Qu et al. [7] proposed the spatial frequency-motivated PCNN model in NSCT domain, and the spatial frequency was used to implement the firing mapping in the method. Liao et al. [8] proposed a shearlet-based fusion method by employing the statistical information of the shearlet coefficients. Considering the fusion procedure of the aforementioned methods, it is obvious that the fusion results are highly determined by the performance of the decomposition abilities.

From the viewpoint of fusion rules, the fusion procedure of the multi-scale geometric analysis theory-based methods can be modeled to be the classification problem of the multi-scale transformation coefficients. There are also three typical categories for the fusion rules: the active level metric-based rule, the kernel learning-based fusion rule, and the neural network-based fusion rule. For the former, the algebraic operations, such as averaging and maxing, are popularly used. The second one includes the ICA, SVM, and PCA. In literature [9], the principal component analysis (PCA) is implemented in dictionary training to reduce the dimension of transform coefficients. In literature [10], the cartoon components and the texture components are combined by ICA. The artifacts are easy to produce for the classification determined by the simple computations for the single coefficient. The neural network-based rule has been popularly reported in recent years, and some good results have been obtained. For example, in literature [11], the PCNN model is used to be the fusion rule by combining the NSCT together. Though good results have been obtained, these models are not abstract enough, which means the features are all in low levels. So, advanced neural network models should be further developed.

Compared with the traditional neural networks, the deeply coupled neural network is the breakthrough in this domain and has been popularly applied in image denoising, image recognition and classification, and in image fusion [12]. For example, a novel image fusion method for the multi-focus image was proposed based on the support value-motivated deep convolutional neural network model in literature [13]; a general multi-focus image fusion framework, called IF-CNN, was developed based on the deeply convolutional neural network in literature [14]. The MFF-GAN and Pan-GAN, under the mechanism of the unsupervised generative adversarial network, are proposed in the literature [15] and [16], respectively, and detail preserving adversarial learning model is proposed in literature [17, 18]. Specially, according to some recent references, good results are always obtained by the GAN-based methods. The main reason lies in its unique characteristics: firstly, the GAN model has more complex and deep network structure than the commonly used neural network; secondly, modeling the fusion process into the adversarial learning is more in line with the general principle of human understanding of the world. However, the common characteristics of the well-known methods are that they are directly developed in the pixel level, the important image features are not carefully used.

In order to overcome the shortcomings of the above methods, a multi-focus image fusion method based on the GAN in the complex shearlet domain is developed. Different from the traditional transformation methods, such as the curvelet and the contourlet, the complex shearlet can divide the source images into high-pass and low-pass sub-bands to provide more useful features. Besides, the computational efficiency of complex shearlet is higher than that of the NSCT to get the same shift invariance. With the help of the GAN, the whole fusion can be modeled by the adversarial learning of the features in the complex shearlet. Therefore, better fusion results can be obtained in the feature level.

The rest of this paper is organized as follows. The details of the whole method are given in Section 2. Experimental results and some important discussions are given in Section 3. Finally, the paper is concluded in Section 4.

2. Methodology

Figure 1 shows the structure of the proposed method. Firstly, the images to be fused are input into the GAN, and at the same time, the complex shearlet is implemented to get the high-pass subbands for them. Then, the features in the complex shearlet domain are computed to produce the new form of the loss function. The loss function is updated to drive the training of the GAN, and the final fusion results can be obtained after the training is finished.

2.1. The Complex Shearlet Transform. As one of the most famous multi-scale geometric transformation tools, the complex shearlet transform can extract directional information of different scales and deliver highly sparse approximations of the 2D signals. Generally speaking, it divides the source images into low-pass and high-pass sub-band images in different levels, i.e., the approximately sparse representation of the source images and the obvious feature information of the images. Different from the discrete wavelet, contourlet, and shearlet, the complex shearlet is realized based on the multi-scale pyramid filters and the Hilbert transform [19, 20]. The former gives the multiple partitions of the image, and the latter provides the directional sub-bands in the complex space. Figure 2 gives an example of the complex shearlet transform on a "clock with the left focus."

2.2. The Feature in the High-Pass Sub-Bands. After the complex shearlet transform is done, the shearlet coefficients with large absolute values are considered to be the sharp brightness or salient features, meaning that they are the focused regions in the source images. Considering the aim of bringing the focus to the fused image, it needs to extract the focus firstly by using the complex shearlet coefficients.

On the other hand, the features in the multi-focus images can be uniformly described by the activity-level

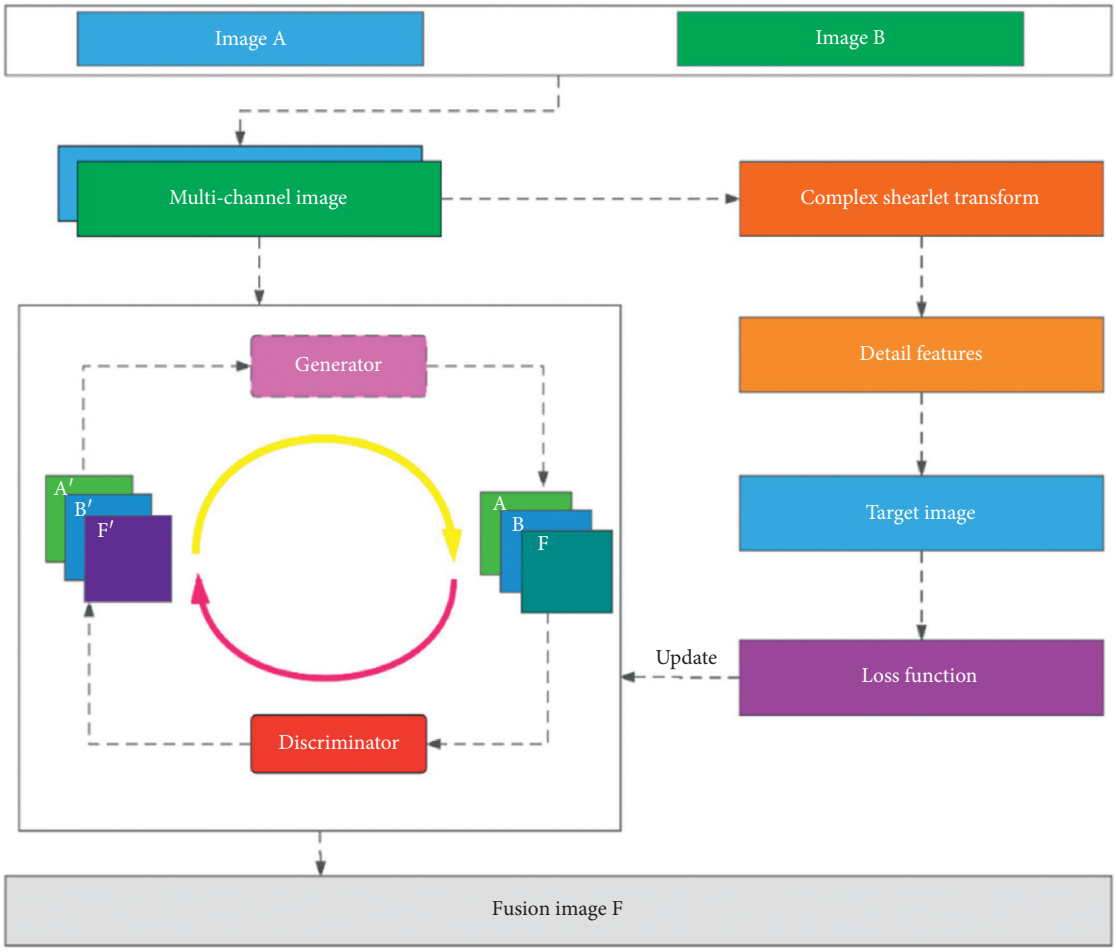


Figure 1: The architecture of the proposed method.

measurements, such as the local energy, standard deviation, and spatial frequency [21, 22]. For the above reasons, local energy and spatial frequency are used to represent the important features in the high-pass coefficients. Furthermore, different from their common form used in other literatures, they are computed in multiple scales and directions.

2.3. The GAN Model

2.3.1. The Structure of the GAN. Usually, the complete structure of the GAN network consists of two parts: the generator and the discriminator [23, 24]. The detailed structure of the GAN model used in this paper is shown in Figures 3 and 4.

For the generator, five convolutional layers are used to extract features. A 5 x 5 convolution kernel is used in the first convolutional layer, and a 3 x 3 convolution kernel is used for the other four layers. The inputs of each layer are connected by the outputs of all the previous layers, with the aim of speeding up the convergence and improving the stability of the model [25] All the activation functions are set to be "ReLU," i.e., rectified linear unit. Furthermore, layer normalization (BN) is also employed to preserve the contrast information of the source images. It calculates the average value of all the dimensional inputs in each layer and finally implements the normalization operation. The advantages are to reduce the sensitivity of initializing data and effectively avoid the gradient disappearance problem [26].

Different from the generator, the main propose of the discriminator is to give the decision by classification. As shown in Figure 4, the discriminator has the same structure with the convolutional neural network which has two inputs, i.e., the Laplacian joint enhanced image from the source images and the fused image from the generator. Four layers of the 3×3 filters are designed to implement the convolution to capture the feature information. Meanwhile, in order to reduce the loss of important information caused by using the downsampling scheme, the activation function is set to be "ReLU." Finally, the fully connected layer is used to classify, and the sigmoid function is employed to output the final results.

2.3.2. The Loss Function. The loss function plays the role of minimizing the loss of the training to get the ideal model, and it usually consists of the generator loss function and discriminator loss function, as shown in the following formula:

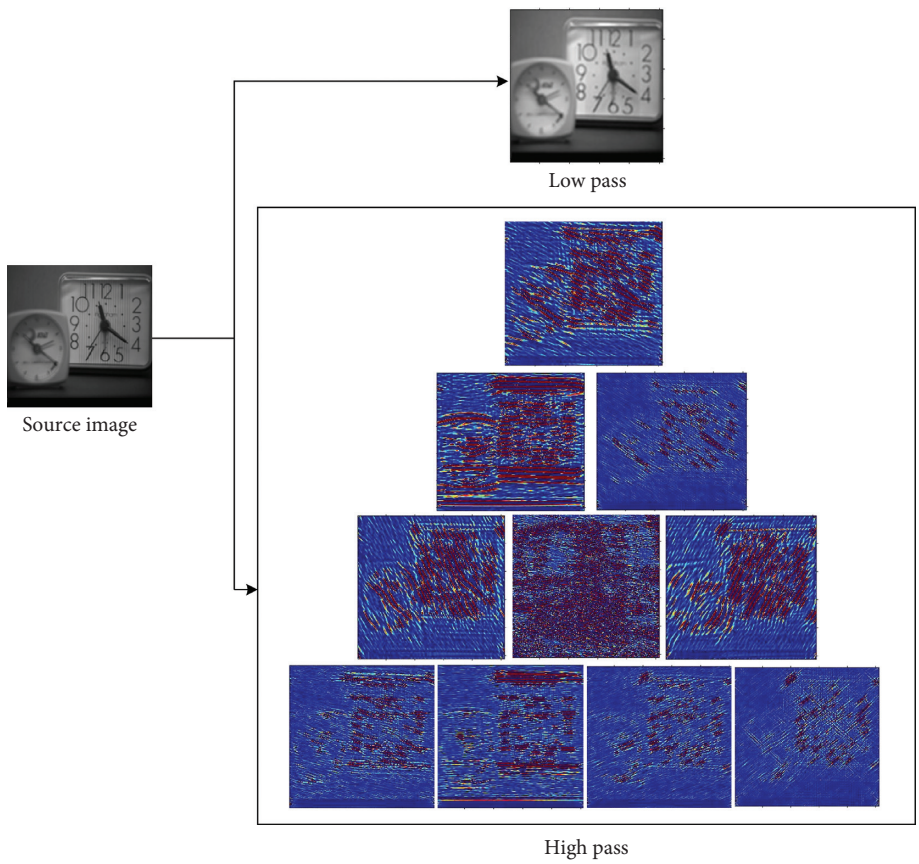


Figure 2: An example of the complex shearlet transformation.

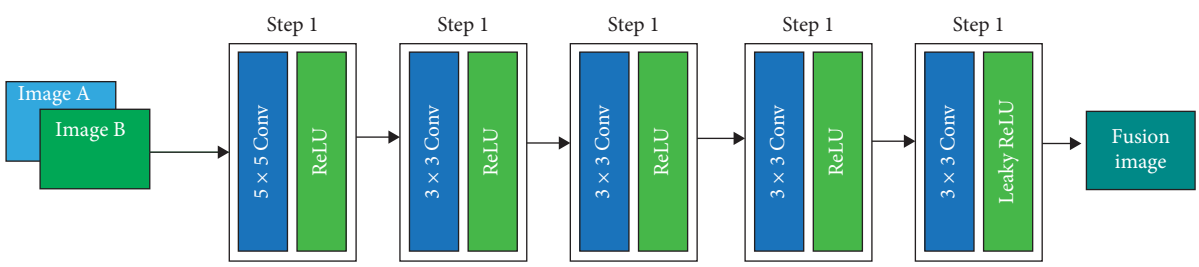


Figure 3: The structure of the generator in GAN.

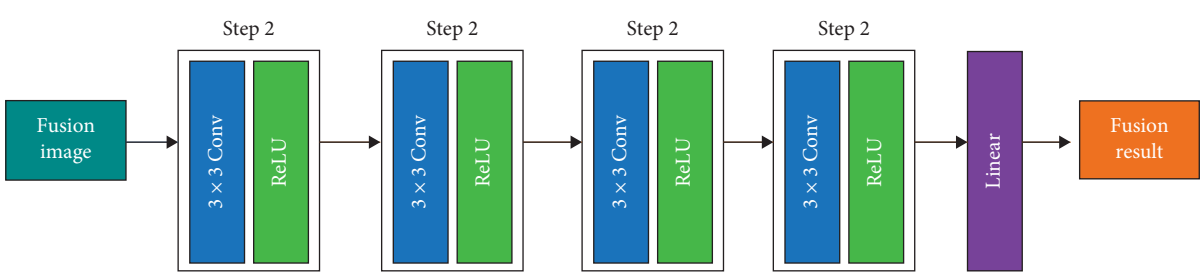


Figure 4: The structure of the discriminator in GAN.

$$L_{\mathrm{GAN}} = \{\min(L_G), \min(L_D)\}, \tag{1}$$

where L_G, L_D are the generator loss function and the discriminator loss function, respectively.

According to the original model, L_G is defined by formula (2). It is computed by the summarizing the confrontation loss R and the content loss C from the procedure of the image generation.

$$L_G = R + \lambda C, \tag{2}$$

$$R = \frac{1}{N}\sum_{n=1}^{N}\left(D\left(I_f^n - \alpha\right)\right)^2, \tag{3}$$

$$C = \mu_1 f_{\mathrm{in}} + \mu_2 f_{\mathrm{grad}}, \tag{4}$$

where λ is a balanced weight between R and C, I_f^n is the target image to be fused, N is the number of fused images, $D(I_f^n)$ is the result of classification, α means that the false data is recognized to be true by the discriminator, f_{in} is the intensity loss, f_{grad} is the gradient loss, and μ is the balanced weight.

L_D can be expressed as

$$L_D = \frac{1}{N}\sum_{n=1}^{N}\left[D\left(G_{I_f}\right) - b\right]^2 + \left[D\left(G_{\mathrm{joint}}\right) - c\right]^2, \tag{5}$$

where G_{joint} is the joint Laplace enhanced gradient map, G_{I_f} is the gradient map of the fused image, and b and c are their labels, respectively.

From all the formulas above, it can be seen that target image to be fused is very important in confrontation learning. The common way to get it is to average the images to be fused or let it be initialized by one of the images to be fused. The drawback is that it is far from the final results and should spend much time and resource to get the optimal decision during the confrontation.

Therefore, a new form of the target image is proposed. Let $C_\lambda(a,b)$ be the low-pass sub-band coefficient at position (a,b), $\lambda = A, B$. The low-pass coefficient of the fused image can be obtained by

$$C_F(a,b) = \begin{cases} C_A(a,b), & E_A(a,b) \geq E_B(a,b), \\ C_B(a,b), & E_A(a,b) < E_B(a,b), \end{cases} \tag{6}$$

where E_λ is the local energy computed in the 3×3 neighborhood.

Let $C_\lambda^{l,k}(i,j)$ be the high-pass coefficient at (i,j) in the l-th sub-band and the k-th level after implementing the complex shearlet transformation, $\lambda = A, B$; then, the high-pass coefficient of the fused image can be obtained by

$$C_F^{l,k}(i,j) = \begin{cases} C_A^{l,k}(i,j), & S_A^{l,k}(i,j) \geq S_B^{l,k}(i,j), \\ C_B^{l,k}(i,j), & S_A^{l,k}(i,j) < S_B^{l,k}(i,j), \end{cases} \tag{7}$$

where s_λ is the spatial frequency which can be computed by the following formula:

$$s_\lambda = \sum_{q=1, q\neq k}^{K}\sum_{p=1, l\neq p}^{L} S\left(C_\lambda^{l,k}, C_\lambda^{p,q}\right). \tag{8}$$

Then, the target image I_f^n can be obtained by applying the inversion of the complex shearlet transform on the fused low-pass and high-pass sub-bands.

3. Experimental Results and Analysis

The experiments are implemented to show the performance of the proposed method. The platform used is Inspur big data server NF5280M4 with Intel Xeon CPU and 256 GB RAM. 100 pairs of multi-focus images are used for the training. All the data can be downloaded from the web [27–30].

Seven typical methods, i.e., PCNN-based method (PCNN for short) [31], the contourlet-based method (contourlet for short) [32], the GAN-based method (GAN for short) [17], the DCNN-based method (DCNN for short) [33], the discrete shearlet-based method (shearlet for short) [34], the convolutional sparse representation-based method (CSR for short) [35], and sparse representation and sum modified-Laplacian-based method (SR-SML for short) [36], are implemented. The level of the complex shearlet is four.

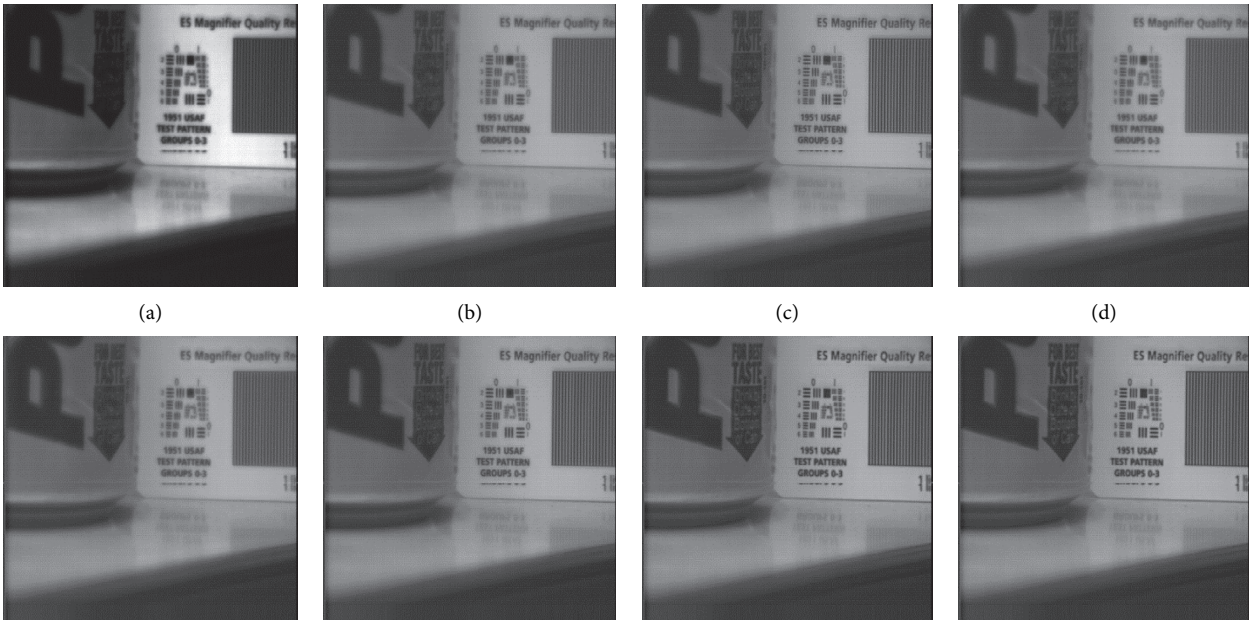
So far, how to evaluate the quality of the fusion results is still a confusing question. Subjectively visual and objectively quantitative comparison is the mainstream practice in this domain. Without loss of generality, mutual information, entropy, standard deviation (MI, En, and SD for short, respectively), and $Q^{\mathrm{AB}/F}$ are selected to be the metrics. The greater their value, the better the fusion images [37–39].

To save space, only "Pepsi-Cola," "Plane," "Clocks," "Flower," "Cup," and "Calendar" are shown in Figure 5. All the fusion results are shown in Figures 6–11 . In Figure 7, the middle part of the "Plane" is partially enlarged to compare the local detail features.

From the above methods, we can see that though the focus regions are expressed better than the source image, the fusion results are different from each other. For the PCNN-based method, blurred edges obviously occur, and so the details are not clear enough. For the contourlet method, shearlet method, CSR method, and SR-SML method, though the results are improved, the contours are sharpened and the phenomenon of ghosting occurs. This can be explained by comparing the ability of the sparse representations for the important image features.

As for the GAN based the DCNN-based method, the results are much clear, but the texture information is not good enough by comparing the results obtained by the proposed method. This is because these two models are directly learned based on the pixel of the images to be fused. The importance of the feature in the procedure of learning is not fully considered. On the other hand, the texture information in the proposed method is highly improved and the ghosting phenomenon is suppressed to the greatest extent. Furthermore, this can also be proved by the enlarged images in Figures 7 and 11. In addition, from the objective comparison in Tables 1 and 2, the best value of the four

Figure 5: Parts of the source images in experiments.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6: The fusion results of "Pepsi-Cola." (a) PCNN. (b) Contourlet. (c) GAN. (d) Shearlet. (e) CSR. (f) SR-SML. (g) DCNN. (h) Proposed.

(a) (b) (c) (d)

Figure 7: Continued.

(e) (f) (g) (h)

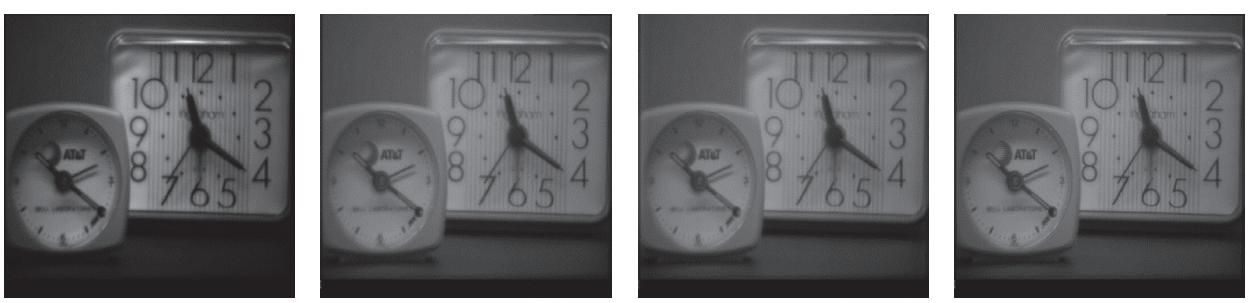
Figure 7: The fusion results of "Plane." (a) PCNN. (b) Contourlet. (c) GAN. (d) Shearlet. (e) CSR. (f) SR-SML. (g) DCNN. (h) Proposed.

Table 1: The fusion performance of the seven methods and the proposed method from Figures 6–8.

Method	Pepsi-Cola				Clock				Plane			
	SD	$Q^{AB/F}$	En	MI	SD	$Q^{AB/F}$	En	MI	SD	$Q^{AB/F}$	En	MI
PCNN	43.71	0.36	6.58	4.46	39.27	0.50	6.98	5.01	44.97	0.40	3.91	3.39
Contourlet	44.11	0.59	7.00	5.39	39.41	0.57	7.00	5.19	46.24	0.48	3.98	3.35
Shearlet	44.07	0.61	7.10	5.34	39.44	0.52	7.00	5.21	46.70	0.49	4.06	3.41
GAN	44.25	0.70	7.16	5.38	39.95	0.62	7.06	5.20	45.842	0.69	4.04	**3.72**
CSR	45.23	0.76	7.10	5.50	40.50	0.68	7.03	5.42	48.10	0.73	4.08	3.60
SR-SML	45.40	**0.78**	7.11	5.55	40.88	**0.69**	7.05	5.44	48.90	**0.76**	4.19	3.62
DCNN	44.80	0.74	7.06	5.41	39.66	0.65	7.00	5.32	46.85	0.68	4.03	3.58
Proposed	**45.25**	**0.78**	**7.20**	**5.60**	**40.88**	**0.69**	**7.10**	**5.53**	**50.15**	**0.76**	**4.28**	**3.72**

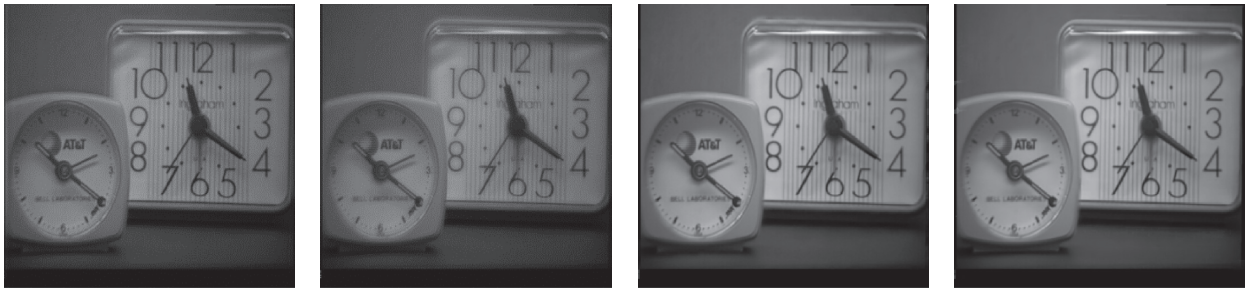
Table 2: The fusion performance of the seven methods and the proposed method from Figures 9–11.

Method	Flower				Cup				Calendar			
	SD	$Q^{AB/F}$	En	MI	SD	$Q^{AB/F}$	En	MI	SD	$Q^{AB/F}$	En	MI
PCNN	40.61	0.41	3.90	4.72	38.22	0.43	4.01	4.21	37.42	0.40	4.88	4.31
Contourlet	40.62	0.40	4.02	4.73	38.39	0.42	4.03	4.38	37.65	0.52	5.35	4.47
Shearlet	40.81	0.42	4.06	4.89	38.44	0.45	4.08	4.41	38.15	0.56	5.55	4.55
GAN	40.81	0.57	4.80	4.99	39.10	0.51	4.10	4.63	38.59	0.60	5.76	4.91
CSR	40.89	0.59	4.65	4.77	38.55	0.54	4.10	4.60	39.03	0.66	5.90	4.89
SR-SML	41.19	0.57	4.66	5.12	38.71	0.55	4.14	4.72	39.11	**0.67**	5.93	4.91
DCNN	40.16	0.60	4.81	5.16	40.01	0.58	4.09	4.81	**39.98**	0.65	5.98	5.24
Proposed	**41.26**	**0.68**	**4.96**	**5.27**	**40.21**	**0.63**	**4.18**	**4.93**	39.91	**0.67**	6.00	**5.29**



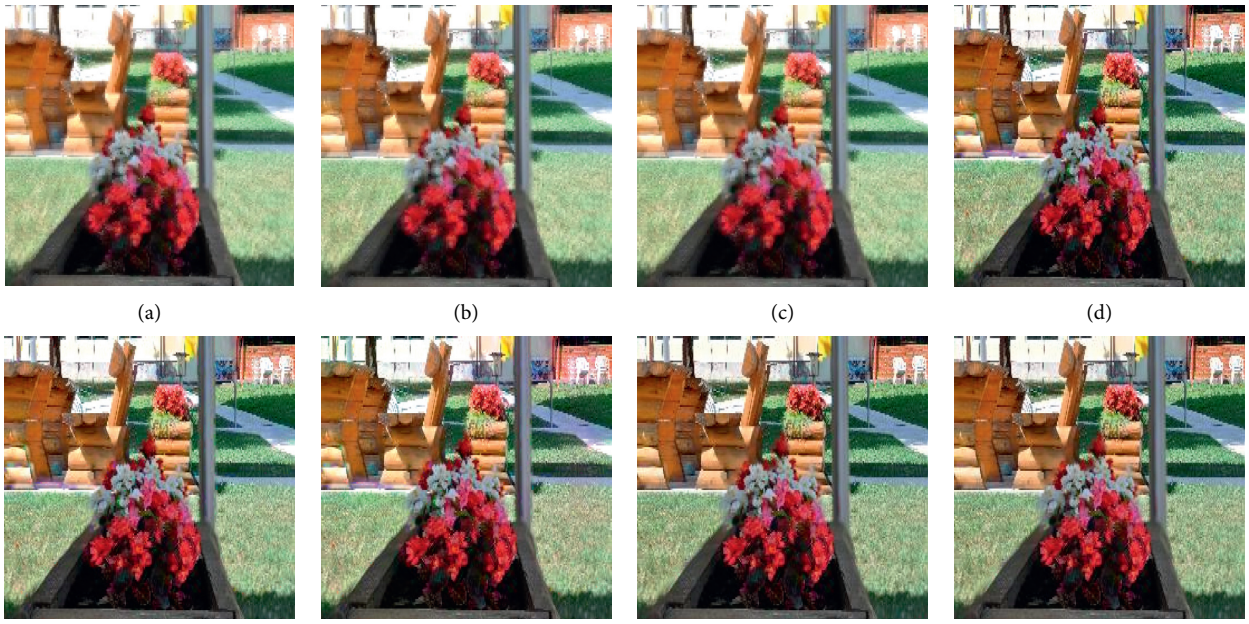
(a) (b) (c) (d)

Figure 8: Continued.



(e) (f) (g) (h)

Figure 8: The fusion results of "Clocks." (a) PCNN. (b) Contourlet. (c) GAN. (d) Shearlet. (e) CSR. (f) SR-SML. (g) DCNN. (h) Proposed.



(a) (b) (c) (d)

(e) (f) (g) (h)

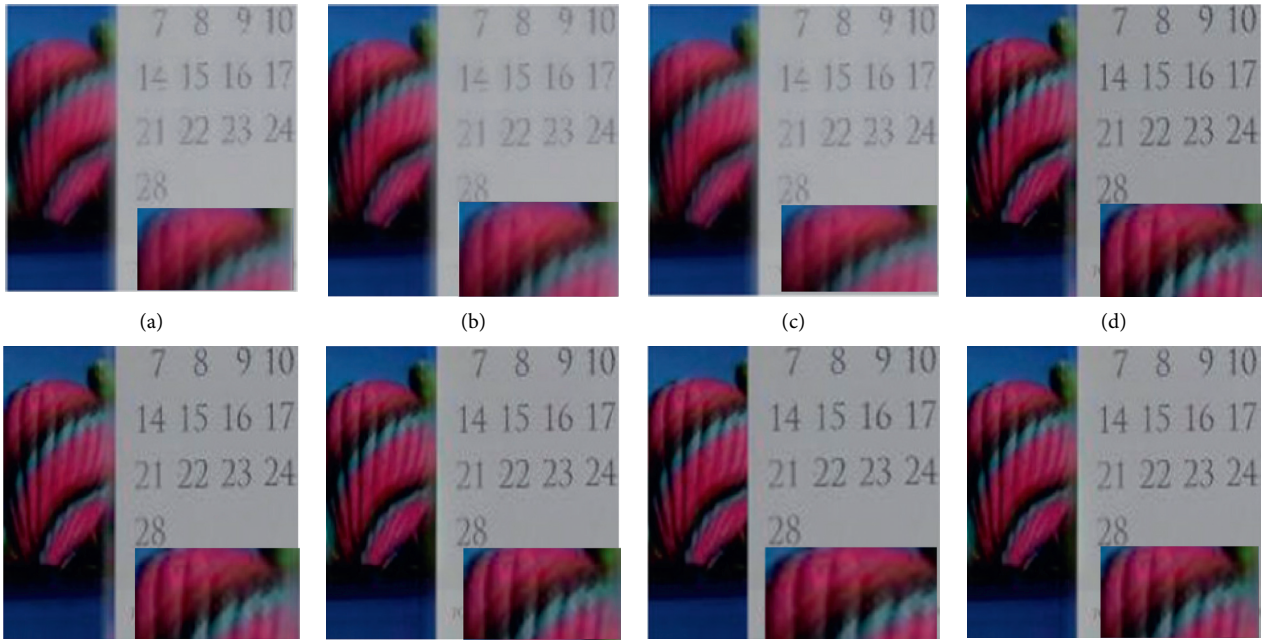
Figure 9: The fusion results of "Flower." (a) PCNN. (b) Contourlet. (c) GAN. (d) Shearlet. (e) CSR. (f) SR-SML. (g) DCNN. (h) Proposed.

(a) (b) (c) (d)

Figure 10: Continued.

(e) (f) (g) (h)

Figure 10: The fusion results of "Cup." (a) PCNN. (b) Contourlet. (c) GAN. (d) Shearlet. (e) CSR. (f) SR-SML. (g) DCNN. (h) Proposed.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 11: The fusion results of "Calendar." (a) PCNN. (b) Contourlet. (c) GAN. (d) Shearlet. (e) CSR. (f) SR-SML. (g) DCNN. (h) Proposed.

metrics can be almost obtained by the proposed method. All the above facts fully demonstrate the effectiveness and accuracy of the proposed method.

4. Conclusion

To get better fusion results for the multi-focus images, the features-motivated generative adversarial network is constructed with the help of the complex shearlet transform. Six typical experiments have been carefully implemented to show the full evidence of the effectiveness and accuracy. In the future, more complex models will be built to further improve the fusion performance.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61502282 and 61902222), the Natural Science Foundation of Shandong Province (ZR2015FQ005), and the Taishan Scholars Program of Shandong Province (tsqn201909109).

References

[1] S. C. Kulkarni and P. P. Rege, "Pixel level fusion techniques for SAR and optical images: a review," *Information Fusion*, vol. 59, pp. 13–29, 2020.

[2] Y. Liu, X. Chen, Z. Wang, Z. J. Wang, R. K. Ward, and X. Wang, "Deep learning for pixel-level image fusion: recent advances and future prospects," *Information Fusion*, vol. 42, pp. 158–173, 2018.

[3] J. Tian, G. Liu, and J. Liu, "Multi-focus image fusion based on edges and focused region extraction," *Optik*, vol. 171, pp. 611–624, 2018.

[4] J. Dou, Q. Qin, and Z. Tu, "Image fusion based on wavelet transform with genetic algorithms and human visual system," *Multimedia Tools and Applications*, vol. 78, no. 9, pp. 12491–12517, 2019.

[5] S. Li and B. Yang, "Multifocus image fusion by combining curvelet and wavelet transform," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1295–1301, 2008.

[6] K. He, D. Zhou, X. Zhang, and R. Nie, "Multi-focus: focused region finding and multi-scale transform for image fusion," *Neurocomputing*, vol. 320, pp. 157–170, 2018.

[7] X.-B. Qu, J.-W. Yan, H.-Z. Xiao, and Z.-Q. Zhu, "Image fusion algorithm based on spatia frequency-motivated pulse coupled neural networks in non-subsampled contourlet transform domain," *Acta Automatica Sinica*, vol. 34, no. 12, pp. 1508–1514, 2009.

[8] Y. Liao, W. Huang, L. Shang et al., "Image fusion based on shearlet and improved PCNN," *Computer Engineering and Application*, vol. 50, pp. 142–146, 2014.

[9] V. P. S. Naidu, "Hybrid DDCT-PCA based multi sensor image fusion," *Journal of Optics*, vol. 43, no. 1, pp. 48–61, 2014.

[10] D. Carone, G. W. J. Harston, J. Garrard et al., "ICA-based denoising for ASL perfusion imaging," *NeuroImage*, vol. 200, pp. 363–372, 2019.

[11] T. Li and Y. Wang, "Biological image fusion using a NSCT based variable weight method," *Information Fusion*, vol. 85, 2011.

[12] Y. Yang, Z. Nie, S. Huang, P. Lin, and J. Wu, "Multilevel features convolutional neural network for multifocus image fusion," *IEEE Transactions on Computational Imaging*, vol. 5, no. 2, pp. 262–273, 2019.

[13] C. Du, S. Gao, Y. Liu, and B. Gao, "Multi-focus image fusion using deep support value convolutional neural network," *Optik*, vol. 176, pp. 567–578, 2019.

[14] Y. Zhang, Y. Liu, P. Sun, H. Yan, X. Zhao, and L. Zhang, "IFCNN: a general image fusion framework based on convolutional neural network," *Information Fusion*, vol. 54, pp. 99–118, 2020.

[15] H. Zhang, Z. Le, Z. Shao, H. Xu, and J. Ma, "MFF-GAN: an unsupervised generative adversarial network with adaptive and gradient joint constraints for multi-focus image fusion," *Information Fusion*, vol. 66, pp. 40–53, 2021.

[16] J. Ma, W. Yu, C. Chen, P. Liang, X. Guo, and J. Jiang, "Pan-GAN: an unsupervised pan-sharpening method for remote sensing image fusion," *Information Fusion*, vol. 62, pp. 110–120, 2020.

[17] J. Ma, W. Yu, P. Liang, C. Li, and J. Jiang, "Fusiongan: a generative adversarial network for infrared and visible image fusion," *Information Fusion*, vol. 48, pp. 11–26, 2019.

[18] B. Lu, Y. Hu, L. Lin et al., "Using ensemble deep learning method to integrate multi-source data to develop national visibility grid data," *Advances in Meteorological Science and Technology*, vol. 8, pp. 77–82, 2018.

[19] X. Liu, Ke Xu, P. Zhou, and J. Chi, "Edge detection of retinal OCT image based on complex shearlet transform," *IET Image Processing*, vol. 13, pp. 1686–1693, 2019.

[20] H. Karbalaali, A. Javaherian, S. Dahlke, and S. Torabi, "Channel edge detection using 2D complex shearlet transform: a case study from the South Caspian Sea," *Exploration Geophysics*, vol. 49, no. 5, pp. 704–712, 2018.

[21] X. Jin, D. Zhou, S. Yao et al., "Multi-focus image fusion method using S-PCNN optimized by particle swarm optimization," *Soft Computing*, vol. 22, no. 19, pp. 6395–6407, 2018.

[22] N. Hayat and M. Imran, "Ghost-free multi exposure image fusion technique using dense SIFT descriptor and guided filter," *Journal of Visual Communication and Image Representation*, vol. 62, pp. 295–308, 2019.

[23] X. Liu, Y. Wang, and Q. Liu, "PSGAN: a generative adversarial network for remote sensing image pan-sharpening," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 873–877, Athens, Greece, October 2018.

[24] X. Guo, R. Nie, J. Cao D, Z. L. Mei, and K. He, "Fusegan: learning to fuse multi-focus image via conditional generative adversarial network," *IEEE Trans. on Multimedia*, vol. 21, pp. 1982–1996, 2019.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Honolulu, HI, USA, July 2017.

[26] C.-B. Du and S.-S. Gao, "Multi-focus image fusion with the all convolutional neural network," *Optoelectronics Letters*, vol. 14, no. 1, pp. 71–75, 2018.

[27] S. Slavica, "Multi-focus image fusion based on empirical mode decomposition," in *Twentieth International Electro technical and Computer Science Conference, ERK 2011*, San Francisco, CA, USA, October 2011.

[28] M. Nejati, S. Samavi, and S. Shirani, "Multi-focus image fusion using dictionary-based sparse representation," *Information Fusion*, vol. 25, pp. 72–84, 2015.

[29] G. Piella, "A general framework for multiresolution image fusion: from pixels to regions," *Information Fusion*, vol. 4, no. 4, pp. 259–280, 2003.

[30] "Multi-focus-Image-Fusion-Dataset," https://github.com/sametaymaz/Multi-focus-Image-Fusion-Dataset.

[31] Z. Wang, Y. Ma, and J. Gu, "Multi-focus image fusion using PCNN," *Pattern Recognition*, vol. 43, no. 6, pp. 2003–2016, 2010.

[32] Z. Zhu, H. Yin, Y. Chai, Y. Li, and G. Qi, "A novel multi-modality image fusion method based on image decomposition and sparse representation," *Information Sciences*, vol. 432, pp. 516–529, 2018.

[33] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Information Fusion*, vol. 36, pp. 191–207, 2017.

[34] X. Liu, Y. Zhou, and J. Wang, "Image fusion based on shearlet transform and regional features," *AEU - International Journal of Electronics and Communications*, vol. 68, no. 6, pp. 471–477, 2014.

[35] Y. Liu, X. Chen, R. K. Ward, and Z. Jane Wang, "Image fusion with convolutional sparse representation," *IEEE Signal Processing Letters*, vol. 23, no. 12, pp. 1882–1886, 2016.

[36] X. Li, X. Zhang, and M. Ding, "A sum-modified-Laplacian and sparse representation based multimodal medical image fusion in Laplacian pyramid domain," *Medical & Biological Engineering & Computing*, vol. 57, no. 10, pp. 2265–2275, 2019.

[37] M. B. A. Haghighat, A. Aghagolzadeh, and H. Seyedarabi, "A non-reference image fusion metric based on mutual information of image features," *Computers & Electrical Engineering*, vol. 37, no. 5, pp. 744–756, 2011.

[38] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

[39] V. Aslantas and E. Bendes, "A new image quality metric for image fusion: the sum of the correlations of differences," *AEU - International Journal of Electronics and Communications*, vol. 69, no. 12, pp. 1890–1896, 2015.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9984659, 12 pages
https://doi.org/10.1155/2021/9984659

Research Article

Route Selection of Multimodal Transport Based on China Railway Transportation

Hui Zhang,[1,2,3,4] **Yao Li**,[1] **Qingpeng Zhang**,[5] **and Dingjun Chen**[1,2,3]

[1]*School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 610031, China*
[2]*National and Local Joint Engineering Laboratory of Comprehensive Intelligent Transportation, Southwest Jiaotong University, Chengdu 610031, China*
[3]*National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Chengdu 610031, China*
[4]*State Laboratory of Rail Transit (Under Preparation), Beijing, China*
[5]*School of Data Science, City University of Hong Kong, Kowloon Tong, Hong Kong*

Correspondence should be addressed to Dingjun Chen; chen-dingjun@163.com

Received 4 April 2021; Revised 25 May 2021; Accepted 30 June 2021; Published 12 July 2021

Academic Editor: Chi-Hua Chen

The advantage of multimodal transport is that it can deliver the goods to their destination in a reasonable combination of transport modes while ensuring security and punctuality. Multimodal transportation can effectively reduce logistics costs, improve logistics efficiency, and reduce environmental pollution. In the process of multimodal transportation, due to the interference of natural factors (weather, terrain, etc.) and some special human factors, it may have different degrees of impact on the transportation time and transportation safety of different transportation modes. Therefore, when choosing a transportation method, it is necessary to consider the transportation time and transportation safety under the interference. However, the current research on multimodal transport has not considered the impact of external interference on transportation time and transportation safety. Compared with other modes of transportation, external interference has a relatively small impact on railway transportation. Railways can safely deliver goods to their destinations on time. Under the background of China's huge railway network and advanced heavy-duty technology, this paper establishes a multimodal transport route selection model for considering railway as the core, introduces time penalty cost and damage compensation cost, and takes the lowest comprehensive transportation cost as the model objective under the premise of considering transportation reliability and transportation safety. Finally, taking a multimodal transport network in China as an example, an improved ant colony algorithm is used to solve the model and the results verify the rationality of the model.

1. Introduction

Multimodal transport [1] is a mode of cargo transportation through the connection and transshipment of two or more transportation modes (such as railway transportation, water transportation, road transportation, and air transportation) to complete the transportation process. At present, China is at an important stage of economic structural transformation and upgrading, and the freight transportation market is facing historical changes. The establishment of a new and efficient logistics channel based on multimodal transport can connect China's inland economy and coastal economy, which is the key to the transformation and upgrading of China's economic structure. This shows that promoting the development of multimodal transport is of great significance. According to statistics, in 2017, China's multimodal freight volume was 1.368 billion tons, accounting for 2.9% of the total freight volume of the society. According to the "Notice on Further Encouraging Multimodal Transport Work" by 18 departments including the Ministry of Transport, China's multimodal transport freight volume will reach 3.02 billion tons in 2020. By then, the scale of multimodal transport freight will account for 6% of the total freight volume%, the proportion is still low. The multimodal transport freight volume accounts for a small proportion of the whole society's freight volume, which is caused by the

incomplete construction of China's current integrated transportation system. With the increasing market requirements, more efficient, more resource-efficient, and more intensive transportation services are required to achieve a rational division of labor and organic linkage of various modes of transportation, and to give full play to the overall efficiency and effectiveness of the integrated transportation system. However, at present, all kinds of transportation modes in the integrated transportation system are self-contained, which are only physically connected through hubs, and the limited joint command is carry out under the emergency condition. There are some problems, such as imperfect mechanism, market segmentation, prominent transport structural contradictions, weak hub service capacity, poor operational coordination, and low safety and service level, which make the great potential of integrated transportation not fully released. It affects the full play of the efficiency of various modes of transport and the public's sense of gain (quoted: the 13th Five-Year Plan of Integrated Transport Services), which cannot meet the economic, reliable, and efficient freight demand.

Compared with other modes of transportation, railway transportation is more economical and environmentally friendly, and the advantages of transportation reliability and transportation security are more obvious. At the same time, railway transportation as a participant of multimodal transportation has good transportation conditions. Data show that in 2018, the national railways completed 3.19 billion tons of cargo shipments, a year-on-year increase of 9.3%, which increased transportation by 272 million tons. With the formation of the "four vertical and four horizontals" networks of high-speed railways and the promotion of the "eight vertical and eight horizontals" high-speed rail network, the operating mileage of the railway reached 131,000 kilometers, of which the operating mileage of the high-speed railway was 31,000 kilometers. It is concluded that China has a huge railway network, and the operating mileage of railways covers the entire country. Although railways have a nationwide road network and huge transportation capacity, they have not been fully utilized. In 2018, China's highway freight volume was 39.568 billion tons, accounting for 76.79% of the country's total freight; China's railway freight volume was 4.022 billion tons, accounting for 7.81% of the country's total freight. So, making full use of railway capacity has become a key goal for China to promote the development of multimodal transport. In response to this situation, relevant Chinese departments have formulated a "road to rail transportation" policy, and railway management is included in the Ministry of Transport. These policies make it possible for railways to play a backbone role in the integrated transportation system. So, it is urgent to study the cooperative operation theory and key technologies of the railway-based integrated transportation system. This paper will deeply study the optimization of multimodal transport path selection with railway transportation as the main body, and realize the rational division of labor and coordinated development of various transportation modes.

In multimodal transport network design, Christine and Sabine [2] conducted a scenario-based assessment of the future position of intermodal transport within Belgium-based case studies and discussed service network design models for consolidation-based freight transport systems; Demir et al. [3] studied the design problem of green multimodal transport network with uncertain transportation time [4–6] and established a stochastic optimization model that can generate robust transportation plans according to different objectives (cost, time, and carbon emissions); Wang and Meng [7] considered a discrete multimodal transport network design problem and decided whether the network planner needs to establish or extend a path to minimize operating costs. Cheng and Jin [8] constructed the multimodal transportation route selection models with congestion considered under different low-carbon polices; Demir et al. [3] introduced a green intermodal service network design problem with travel time uncertainty (GISND-TTU) for combined offline intermodal routing decisions of multiple commodities. In the path optimization of multimodal transport, many scholars have made in-depth research based on different aspects. Fazayeli et al. [9] studied the problem of multimodal transport path optimization under fuzzy demand, established a mixed-integer fuzzy mathematical model, and solved it by genetic algorithms [10]; Fotuhi studied the optimization problem of multimodal transport path with uncertain network topology [11]; Idri et al. [12] introduced a search method for the shortest path algorithm in parallel distributed architecture to solve the path optimization problem of dynamic multimodal transport network with time dependence [13, 14]; Liu et al. [15] established a super transportation network to predict the generalized cost of multimodal transportation, which provides technical support for the comprehensive transportation planning. Li constructed a route selection model for cargo multimodal transportation and used an improved ant colony algorithm to be solved. Tian et al. [16] explored the influence of multiple time windows on the route selection of fresh products and designed genetic algorithms to solve it. Zhu et al. established an evaluation model of container multimodal transport based on BP neural network and solved the optimal path. Russo et al. [17] research object is a variable of the operating cycle of a rail transit terminal, the variable considered is the overall average time of the terminal operation cycle relative to the number of trucks and vehicles entering the terminal, which solved the problem of the efficiency of public rail transport; Abderrahman et al. [18] proposed a robust optimization model for the transportation cost of multimodal freight and the uncertainty of network terminal capacity [19]; Liu et al. [20] in the context of global containerized transportation of soybeans considers the transportation and purchase costs of soybeans, establishes a multimodal transport network model, and optimizes the soybean transportation path [21]; Verma et al. [22] designed a dual-objective mixed-integer linear programming model for the problem of joint transportation path selection for dangerous goods with minimum transportation costs and risks of transportation and designed a model-solving algorithm based on taboo search; Toumazi and

Kwon [23] introduced the conditional value-at-risk (CVaR) theory and built a CVaR-based road transport route optimization model for dangerous goods under a time-varying network [16, 24]. The optimal dangerous goods road transport can be selected based on the risk aversion of decision makers' path; Sun and Long [25] took into account the customer's need for timeliness of transportation when planning the multimodal transport path [26] and obtained the optimal solution through Pareto optimality.

At present, the research on multimodal transport does not consider the influence of natural factors such as weather, terrain, and some special human factors on the transport [15]. In actual transportation, these factors may lead to the delay [27] of transport time, and even the occurrence of dangerous accidents [28] and damage of goods. At this time, the cost and time of the original optimal path will change, and the optimal path will change accordingly [29, 30]. The multimodal transport research based on railways mostly focuses on transportation policy, and there are few research projects on path selection. Therefore, this paper establishes a multimodal transport route selection model based on railway transportation, introduces time penalty cost and damage compensation cost, considering the transportation reliability and transportation safety factors in the transportation process, aiming at the minimum transportation cost, and finds the optimal path that is in line with reality.

2. Construction of Multimodal Transport Model

2.1. Problem Description. The route selection problem of multimodal transportation mainly based on railway transportation can be described as follows: In a multimodal transport network $G = (V, A)$, V is the node set of networks G and A is the path set of networks G. The shipper requires that a batch of goods be transported from the origination $o \in A$ to the destination $d \in A$. The carrier needs to consider the reliability and safety of the transportation routes between the origination and destination, so as to select the transportation route and mode that can reach the destination within the time specified by the shipper and ensure that the goods are intact.

2.2. Variable Definitions. The notation of parameters and variables is shown in Table 1.

3. Objective Functions

The goal of multimodal transport is to maximize the transport efficiency and spend as little cost and time as possible to make the goods arrive at the destination safely. Multimodal transport is the combined transport of multiple modes of transport, and transportation cost and transportation time for different modes of transport are different. Therefore, when calculating the model target, it is necessary to choose a transportation mode with lower transportation cost and shorter transportation time. However, in the actual transportation process, the transportation cost and transportation time of various transportation modes are not fixed. They will change with the interference of natural factors and some special human factors. Different transportation modes have different antijamming capabilities, and transportation costs and transportation time have different degrees of variation. For example, in foggy weather conditions, various modes of transportation will be affected, but rail transportation will be less affected, and the transportation time will be slightly extended; the speed of road transportation will be greatly reduced, and safety accidents may occur due to operational errors, not only the safety of life is involved but also the transported goods will be damaged. The carrier will have to compensate the shipper for losses, and the transportation cost and transportation time will increase significantly.

When the transportation cost and transportation time are dynamically changing, it will dynamically affect the choice of transportation mode. Therefore, the model built in this paper focuses on the change of transportation cost and transportation time during transportation. In this paper, the degree of change in transportation time is expressed as transportation reliability, and the time penalty cost is established. The degree of change in transportation cost is expressed as transportation safety, and the damage compensation cost is established. At the same time, considering the global low-carbon environmental protection at the current stage, low-carbon transportation is the top priority for the country. Multimodal operation is an emerging efficient transportation mode. It is also necessary to reduce carbon emissions as much as possible in response to national calls. Therefore, this paper will start from the cost, energy conservation, time, transport reliability, and transport safety to build a path optimization model with the minimum comprehensive cost:

$$\min Z = \omega_1 \left(C_1 + C_2 + C_3\right) + \omega_2 C_4 + \omega_3 C_5. \tag{1}$$

In the formula, Z is the comprehensive transport cost; C_1, C_2, C_3, C_4, and C_5 are transportation cost, transit cost, carbon emission cost, time penalty cost, and damage compensation cost, respectively; and ω_1, ω_2, and ω_3 are the weights of each cost in the objective function.

3.1. Transportation Cost and Transit Cost. The cost of freight transportation between adjacent nodes is the transportation cost, and the cost of replacing the transportation mode inside the transferable node is the transit cost:

$$\begin{aligned} \min C_1 &= \sum_{i\in M}\sum_{j\in M}\sum_{s\in S} Q c_{ij}^{s} d_{ij}^{s} x_{ij}^{s}, \\ \min C_2 &= \sum_{i\in M}\sum_{s\in S}\sum_{s'\in S} Q c_{i}^{ss'} y_{i}^{ss'}. \end{aligned} \tag{2}$$

3.2. Carbon Emission Cost. In order to develop a low-carbon economy, countries all over the world have studied energy conservation and emission reduction and proposed a carbon tax scheme. Carbon tax is a tax imposed on carbon emissions

Table 1: Subscripts and parameters used in mathematical formulations.

Symbol	Description
M	Network node set (all nodes including the start node and the end node)
K	Transportable node set
S	$S = \{s, s', \ldots\}$, transportation mode collection
Q	Total quantity of goods to be transported in the multimodal transport model
π	Unit value of goods that need to be transported
ψ_{ij}^{s}	Accident damage coefficient: from node city i to node city j, transport mode$s$is used for transportation
d_{ij}^{s}	Transportation mileage: from node city i to node city j, transport mode$s$is used for transportation
μ_{ij}^{s}	Time delay coefficient: from node city i to node city j, transport mode$s$is used for transportation
c_{ij}^{s}	Unit transportation cost: from node city i to node city$j$, transport mode$s$is used for transportation
$c_{i}^{ss'}$	Unit transfer cost: at the transferable node i, the mode of transport is changed from$s$to s'
$T_{i}^{ss'}$	Transit time: at the transferable node i, the mode of transport is changed from s to s'
V_S	Average speed of transportation mode s
E_S	Carbon emission factor: unit carbon emissions of transportation mode s
E	Total carbon emissions allowed in the model
T	The maximum delivery time of the goods specified in the model
x_{ij}^{s}	Decision variables: if transportation mode$s$is selected from city i to node city j, then the value is 1, otherwise the value is 0
$y_{i}^{ss'}$	Decision variables: if the mode of transportation changes from s to s' in node city i, then the value is 1, otherwise the value is 0
Cd	Cd represents the calculated value of total carbon emissions
X	The total transportation time from o to D
q_{ij}^{s}	From the city i to the city j, the transportation volume of mode s is adopted
q_{hi}^{s}	From the city h to the city i, the transportation volume of mode s is adopted

and is an important means of limiting carbon emissions. The calculation formula of carbon tax is as follows:

$$\theta = \sum_{0}^{t} D_t (1+\lambda)^{-t}. \tag{3}$$

In the formula, D_t is the damage value of the$t$-year caused by the unit carbon emissions and λ is the discount rate. In fact, the carbon tax is not levied as high as possible. If the carbon tax is higher, it will restrict the economic development of the country. If the carbon tax is too low, it will not play a good role in energy conservation and emission reduction. According to the research of experts and scholars in China, the proposal of the Ministry of environmental protection for carbon tax is 20 \$/ton. Therefore, in this paper, the carbon tax is assumed to be 20 \$/ton and 0.02 \$/kg. The carbon emission cost is the product of carbon emissions and carbon tax:

$$\min C_3 = \theta \times \sum_{i\in M} \sum_{j\in M} \sum_{s\in S} Q E_S d_{ij}^{s} x_{ij}^{s}. \tag{4}$$

3.3. Transportation Reliability and Time Penalty Cost. Transport reliability herein refers to the on-time arrival rate from the starting point to the end point within a specified time using either mode of transport. In the actual transportation process, various modes of transportation will be affected by natural factors and some special human factors, resulting in delays in transportation, such as traffic jams caused by road conditions; trains will stop and wait during the meeting, and long ramps will have a great impact on the running speed and so on. Transportation delay will prolong the transportation time and may exceed the delivery time limit, so if the model does not take into account the transportation reliability factors, the obtained optimal path will lose its significance in the actual transportation context. In order to quantify the impact of transportation reliability on transportation time, this paper defines the time delay coefficient: taking into account the complex influencing factors between adjacent nodes and taking values within 0–1 so that the transportation time of the section is extended linearly on the basis of the original transportation time, as shown in Figure 1.

In the diagram, t represents the transportation time between two nodes without considering the delay, and μ is the time delay coefficient. When $\mu = 0$, there is no time delay in the transportation process of goods, and the actual transportation time of goods is the initial transportation time t. When $\mu = 1$, there is time delay in the transportation process of goods, and the actual transportation time is the initial time t plus the delay time quantified as t. When μ is between 0 and 1, the delay time of goods is related to the delay coefficient, defining that the delay time has a linear relationship with the delay coefficient.

A certain penalty is imposed for transportation schemes that exceed the time limit. In order to quantify the impact of the time limit of delivery on the transportation plan selection, the time penalty function is introduced in this paper. Taking into account the changes in the psychological state of the consignee while waiting for the delivery of the goods, this paper adopts the fuzzy convenience function to express and defines 2 hours as a state change gradient in case of exceeding the arrival time limit and assigns different degrees of punishment to different state gradients, as shown in Figure 2.

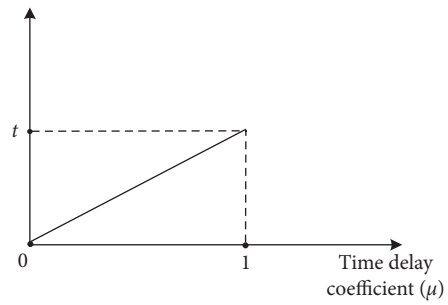


Figure 1: Time delay function.

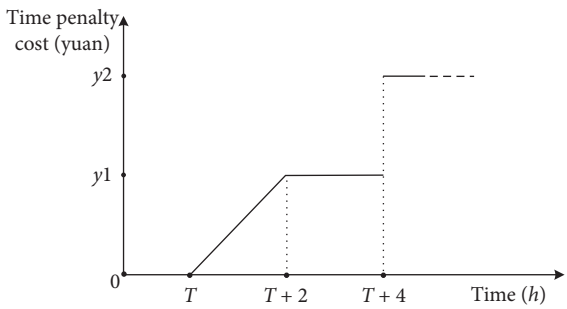


Figure 2: Time penalty cost function.

$$X = \sum_{i\in M}\sum_{j\in M}\sum_{s\in S}\left(1+\mu_{ij}^{s}\right)\frac{d_{ij}^{s}}{V_S}x_{ij}^{s} + \sum_{i\in M}\sum_{s\in S}\sum_{s'\in S}T_i^{ss'}y_i^{ss'}, \quad (5)$$

$$\min C_4 = \begin{cases} 0, & X \le T, \\ \dfrac{y_1}{2}(X-T), & T \le X \le T+2, \\ y_1, & T+2 \le X \le T+4, \\ y_2, & T+4 \le X. \end{cases} \quad (6)$$

Formula (5) is the transportation time calculation formula for any transportation scheme: the first part is the sum of transportation time and delay time, and the second part is the transit time of the transportable node. Formula (6) is the time penalty cost calculation formula, when the transportation time is from T to $T+2$, the penalty function is a linear increasing function; when the time is from $T+2$ to $T+4$, the penalty function is a constant (y_1); when the transportation time exceeds $T+4$, the penalty function is a constant (y_2).

3.4. Transportation Safety and Damage Compensation Costs. Transport safety in this paper refers to the value loss rate of goods transported during transportation from the starting point to the end point. The influencing factors mainly include road conditions and traffic management status, vehicle technical performance, and warranty quality, as well as the driver's operating skill level and responsibility. If a safety accident occurs during transportation, the transportation goods will be damaged. The carrier shall compensate the consignor for the corresponding loss according to the degree of damage to the goods because the liability of the accident is that the carrier has not planned the transportation route. If the safety factors in the transportation process are not considered when optimizing the route, the final result is likely to lose more. In order to facilitate the calculation, this paper assumes that the degree of damage to the goods is determined by the severity of the accident when a safety accident occurs during transportation. That is to say, if the severity of the accident is relatively low, the goods may only be partially damaged rather than completely damaged. In order to quantify the impact of the severity of the accident on the damage of the goods, this paper proposes and defines the accident damage coefficient: the value is within 0–1, and the change from 0 to 1 indicates that the degree of damage of the goods increases gradually and the amount of compensation required by the carrier increases gradually, and the amount of compensation is based on the value of the goods:

$$\min C_5 = \sum_{i\in M}\sum_{j\in M}\sum_{s\in S}\psi_{ij}^{s}x_{ij}^{s}Q\pi. \quad (7)$$

4. Constraints

4.1. Transportation Process Constraint

$$\sum_{s\in S}x_{ij}^{s} \le 1, \qquad \forall (i,j) \in M, \quad (8)$$

$$\sum_{s,s'\in S}y_i^{ss'} \le 1, \qquad \forall i \in K, \quad (9)$$

$$x_{hi}^{s} + x_{ij}^{s'} \ge 2y_i^{ss'}, \quad \forall h,i,j \in M, \forall s,s' \in S, \quad (10)$$

$$\sum_{s\in S}q_{hi}^{s} = \sum_{s\in S}q_{ij}^{s}, \quad \forall (h,i,j) \in M. \quad (11)$$

Constraint (8) makes sure that only one mode of transport can be selected between any adjacent nodes. Constraint (9) ensures that only one transit can occur in any transit node. Constraint (10) guarantees the continuity of transportation mode during transit. Constraint (11) is the flow equilibrium constraint of nodes in the network.

4.2. Carbon Emission Constraint

$$\sum_{i\in M}\sum_{j\in M}\sum_{s\in S}QE_S d_{ij}^{s}x_{ij}^{s} = Cd, \quad (12)$$

$$Cd \le E. \quad (13)$$

Constraint (12) is the formula of total carbon emissions, and for this paper, which is a known constant. Constraint (13) ensures that the transportation plan meets the total carbon emission limit.

5. Multiobjective Ant Colony Algorithm

The ant colony algorithm is a simulation evolution algorithm proposed by DORIGO. It is an effective means to solve the

multiobjective optimization problem in discrete space; however, the basic ant colony algorithm has the problems of slow running efficiency, premature convergence, and easy to fall into a local optimal solution. This paper combines the solution requirements of the multimodal transport path selection model. In order to improve the effectiveness of the algorithm, the basic algorithm is improved accordingly.

5.1. Node Transition Probability. Although the objective function of the model in this paper is the total cost value in the transportation process, the time penalty cost is determined by the length of the transportation time. The model target expectation cost and time are as small as possible, so in the improved ant colony algorithm, a cost heuristic function and a time heuristic function need to be set at the same time. It will mainly rely on the influence of path pheromone concentration, cost heuristic experience, and time heuristic experience when choosing to reach the node:

$$P_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)\theta_{ij}^{\lambda}(t)}{\sum_{s\in \text{allowed}_k}\tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)\theta_{is}^{\lambda}(t)}, & j \in \text{allowed}_k, \\ 0 & \text{otherwise}, \end{cases} \tag{14}$$

$$\eta_{ij} = \frac{1}{\sum_{s\in S}Qc_{ij}^{s}\left(d_{ij}^{s}/V_S\right)x_{ij}^{s} + \sum_{s\in S}\theta QE_S d_{ij}^{s}x_{ij}^{s} + \sum_{s\in S}\psi_{ij}^{s}x_{ij}^{s}Q\pi + \sum_{s\in S}\sum_{s'\in S}Qc_i^{ss'}y_i^{ss'}}, \tag{15}$$

$$\theta_{ij} = \frac{1}{\sum_{s\in S}\left(d_{ij}^{s}/V_S\right)x_{ij}^{s} + \sum_{s\in S}\sum_{s'\in S}T_i^{ss'}y_i^{ss'}}. \tag{16}$$

Formula (14) is the node transition probability formula in the multiobjective ant colony algorithm, where α is the information heuristic factor, β is the cost expectation heuristic factor, λ is the time expectation heuristic factor, and τ_{ij} is the pheromone concentration on path (i, j). Formula (15) is a cost heuristic function, whose value is inversely proportional to the sum of transportation cost, carbon emission cost, and damage compensation cost on path 1 and transit cost of $i$node. Formula (16) is a time heuristic function, whose value is inversely proportional to the sum of transportation time on path (i, j) and transit time of $i$node.

5.2. Pheromone Update Rule. The principle of the positive feedback mechanism of ant colony algorithm is based on the continuous update of pheromone, so the pheromone update rule is the key to solving the problem of speed and accuracy of ant colony algorithm. This paper considers the basic idea of ant colony algorithm with elite strategy: at the end of each cycle, additional pheromone enhancement is given to the optimal solution found in this cycle in order to make the optimal solution found so far more attractive to the ants in the next cycle. The pheromone is updated according to the following formula:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t, t+n) + \Delta\tau_{ij}^{*}(t, t+n), \tag{17}$$

$$\Delta\tau_{ij}^{*}(t, t+n) = \begin{cases} \sigma \cdot \dfrac{q}{C^{*}}, & \text{If path } (i, j) \text{ is part of the optimal path in this cycle}, \\ 0, & \text{otherwise}. \end{cases} \tag{18}$$

Formula (17) is the pheromone concentration on path (i, j) at time $t+n$, and ρ is the pheromone global volatilization coefficient, $\Delta\tau_{ij}(t, t+n)$ is the amount of pheromone increase on path (i, j) from time t to time $t+n$. Formula (18) is the amount of pheromone increase on path (i, j) caused by elite ants from time t to time $t+n$, q is the pheromone intensity, σ is the number of elite ants, and C^{*} is the optimal solution value found in this cycle.

Using the elite strategy can make the ant system find a better solution through fewer cycles, and the solution speed is faster. However, after using the elite strategy, the attractiveness of the current optimal solution increases, so that the search process is concentrated near the optimal solution found so far, thereby preventing further searches for a better solution. The problem of premature convergence is more likely to occur. In order to maintain the selection pressure, this paper extends the concept of ranking in genetic

algorithms to the elite mechanism and calculates $\Delta\tau_{ij}(t, t+n)$ by a weighted method. After completing a cycle, the ants are sorted according to the size of the target value found in the cycle, and the ant's contribution to the pheromone update is considered to be inversely proportional to the target value of the path the ant passes. The contribution of the ant to the pheromone update is weighted according to the ranking of the ant (ξ). In order to ensure the speed of the solution while expanding the solution space, it is considered that only the top ($m/4$) ants can contribute to the pheromone update, and the amount of pheromone obtained by the path it passes is proportional to the ant's ranking. $\Delta\tau_{ij}(t, t+n)$ is calculated as follows:

$$\Delta\tau_{ij}(t, t+n) = \sum_{\mu=1}^{(m/4)} \Delta\tau_{ij}^{\xi}(t, t+n), \tag{19}$$

$$\Delta\tau_{ij}^{\xi}(t, t+n) = \begin{cases} \dfrac{C_{av} - C_{\xi}}{C_{av} - C^{*}} \cdot \dfrac{q}{C_{\xi}}, & \text{if ant } \xi \text{ passes the path } (i, j), \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

Formula (20) represents the contribution of the ant ξ to the increase in the amount of pheromone on the path (i, j), ξ is the ranking of the ants, C_{av} represents the average value of the target of this cycle, and C_{ξ} represents the target value found by the ant ξ.

5.3. Pheromone Limit Interval. Although the aforementioned improved pheromone update rule can solve the problems of solution speed and precocious convergence, it is easy to fall into the dilemma of local optimization and the algorithm stops searching. Therefore, in this paper, the idea of MMAS is introduced in the process of pheromone updating. The upper and lower limits are imposed on the pheromone, and the limiting interval of the pheromone is set to $[\tau_{\min}, \tau_{\max}]$. When performing pheromone update according to the aforementioned pheromone update rules, the following is obtained:

(1) If $\tau_{ij}(t) > \tau_{\max}$ occurs, $\tau_{ij}(t) = \tau_{\max}$ is required

(2) If $\tau_{ij}(t) < \tau_{\min}$ occurs, $\tau_{ij}(t) = \tau_{\min}$ is required

This can effectively avoid that the amount of pheromone on one path is much larger than the rest, so that all ants are concentrated on the same path, resulting in a local optimal situation.

5.4. Algorithm Flowchart. In summary, the multiobjective ant colony algorithm flowchart is as shown in Figure 3.

6. Case Study

Take China's multimodal transport network with 10 cities as an example, it involves 10 cities, the origination is Chengdu and the destination is Shanghai, as shown in Table 2. Since there are many inland cities in this multimodal transport network and the water transport is poor, we considered three transport modes: road, railway, and air in this case. The node cities in the multimodal transport network are numbered sequentially, and the schematic diagram of the multimodal transport network of this case is made accordingly, as shown in Figure 4.

This case involves three transportation modes. We need to consider how to clearly express the ant's choice behavior for transportation mode when using the ant colony algorithm. Therefore, we have simply dealt with the multimodal transport network. Each transport node is expanded into multiple virtual subnodes according to the number of transport modes. For example, transfer node 3 is extended to railway node, highway node, and airport node. In the processed network diagram, there is a transport problem among multiple subnodes of the same node. But it is not the ant to select the next arrival node, that is to say, there is no sequence between the subnodes, which is the same level parallel relationship. The network deformation diagram is shown in Figure 5.

The transportation mileage, time delay coefficient, and accident damage coefficient of different transportation modes among node cities are shown in Table 3. The time delay coefficient is given according to the road condition of each node city and the different characteristics of transportation mode. The accident damage coefficient is given by the traffic management status of each node city and the technical performance of each transportation mode.

The requirements for freight volume, transportation time limit, and unit value of goods by freight forwarders and the requirements for carbon emission limit by government officials are shown in Table 4.

The unit transit cost and unit transit time between different modes of transport are shown in Table 5.

The transportation cost, average speed, and carbon emission factors of each transportation mode are shown in Table 6.

6.1. Results' Analysis. The multiobjective ant colony algorithm is used to write the algorithm code on the MATLAB R2014a platform. The algorithm parameters are set as follows: $\lambda = 3, \alpha = 1, \beta = 3, q = 2, \rho = 0.9$, the number of ants is 100, and the number of iterations is 200, $y_1 = 10000, y_2 = 20000$. Taking the weight of the objective function as 1/3 as an example, the optimal path of multimodal transport is Chengdu to Chongqing to Wuhan to Hefei to Nanjing, and the total cost is 331750. To explore the results of multimodal transport route optimization considering different cost factors, the weight of each objective function is assigned in turn, and the following conclusions are drawn in this paper.

6.2. Objective Weight Sensitivity Analysis. In order to explore the multimodal path optimization results when considering different cost factors, this paper analyzes the sensitivity of the objective function weights as shown in Table 7 and draws the following conclusions:

In Figure 6, Experiment (1, 5) and (2, 4) are compared, it is found that when the weight of damage compensation cost

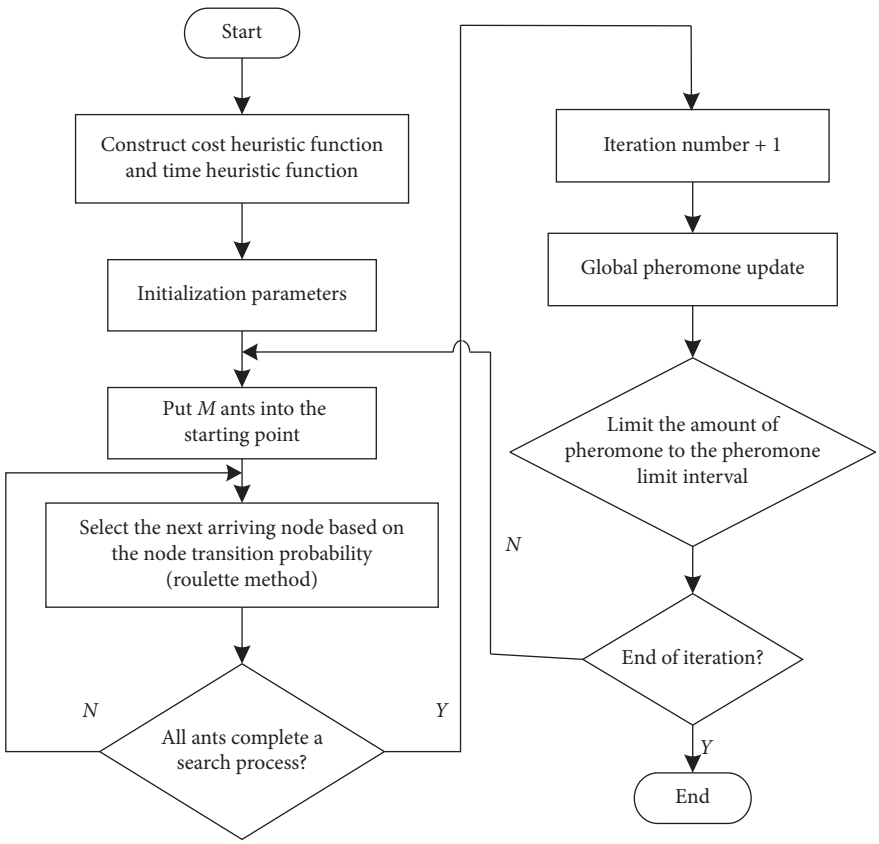


Figure 3: Multiobjective ant colony algorithm flowchart.

Table 2: Node city number.

Number	1	2	3	4	5
Name of node city	Chengdu	Xi'an	Chongqing	Guiyang	Zhengzhou
Number	6	7	8	9	10
Name of node city	Wuhan	Changsha	Hefei	Nanchang	Nanjing

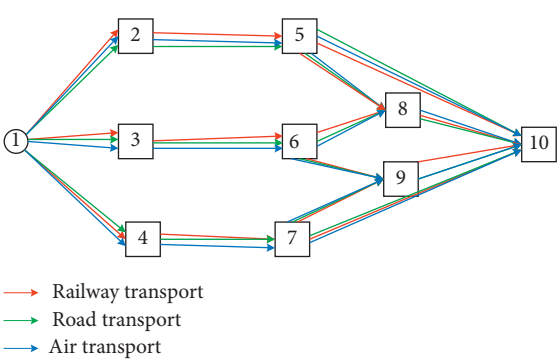


Figure 4: Multimodal network diagram.

is the same, and the weight of the sum of transportation cost, transshipment cost, and carbon emission cost is high, the optimal path is the railway-road combined transportation. The railway-road combined transport pays more attention to the cost of transport. It pays more attention to the size of the cost incurred in the transportation process. When the weight

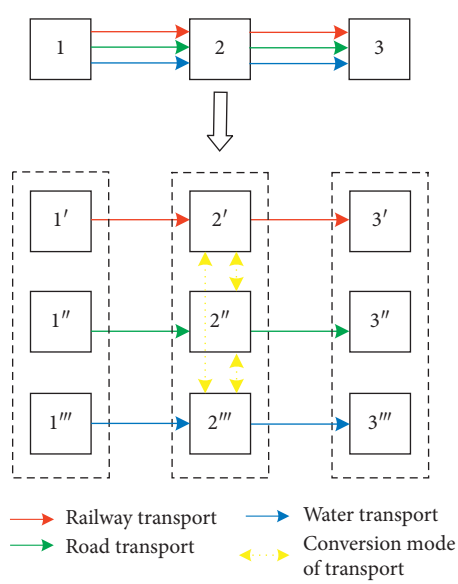


Figure 5: Transit node deformation.

Table 3: Transport distance and time delay coefficient between node pairs.

Adjacent node	(1, 2)	(1, 3)	(1, 4)	(2, 5)	(3, 6)	(4, 7)	(5, 8)
Railway mileage (km)	842	504	967	511	1233	956	645
Time delay coefficient	0.19	0.08	0.12	0.21	0.07	0.16	0.10
Accident damage coefficient	0.04	0.01	0.06	0.08	0.02	0.07	0.03
Highway mileage (km)	719	326	721	479	898	872	673
Time delay coefficient	0.14	0.11	0.09	0.17	0.07	0.13	0.19
Accident damage coefficient	0.09	0.02	0.08	0.05	0.02	0.08	0.13
Waterway mileage (km)	612	270	521	434	752	645	465
Time delay coefficient	0.16	0.12	0.12	0.19	0.09	0.14	0.15
Accident damage coefficient	0.15	0.02	0.12	0.07	0.08	0.05	0.06
Adjacent node	(5, 10)	(6, 8)	(6, 9)	(7, 9)	(7, 10)	(8, 10)	(9, 10)
Railway mileage (km)	695	1181	365	418	1200	312	838
Time delay coefficient	0.18	0.11	0.15	0.13	0.14	0.10	0.17
Accident damage coefficient	0.05	0.03	0.06	0.04	0.05	0.05	0.07
Highway mileage (km)	698	495	370	399	921	178	603
Time delay coefficient	0.15	0.17	0.10	0.16	0.20	0.12	0.18
Accident damage coefficient	0.06	0.05	0.18	0.05	0.14	0.02	0.09
Waterway mileage (km)	564	316	264	290	705	145	468
Time delay coefficient	0.19	0.13	0.16	0.18	0.17	0.11	0.15
Accident damage coefficient	0.04	0.05	0.04	0.05	0.05	0.01	0.17

Table 4: Parameter selection.

Parameter	Freight (t)	Transportation time limit (h)	Carbon emission limit (kg)	Value of goods (yuan/t)
Value	100	45	10000	1000

Table 5: Transit cost and transit time of transit node.

Transit	Rail transport		Road transport		Air transport	
	Cost (yuan/t)	Time (h/t)	Cost (yuan/t)	Time (h/t)	Cost (yuan/t)	Time (h/t)
Rail transport	—	—	12	0.01	12	0.02
Road transport	10	0.01	—	—	15	0.03
Air transport	12	0.02	15	0.03	—	—

Note. The unit of transit cost is yuan/t; and the unit of transit time is h/t.

Table 6: Transport mode parameters.

Mode of transport	Transportation cost (yuan/t·km)	Average speed (km/h)	Unit carbon emissions/(kg/t·km)
Highway	1.5	100	0.12
Railway	1	75	0.025
Waterway	5	600	1.05

Table 7: Weight assignment experiment number.

Number	1			2			3			4			5		
Weight assignment	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
	0.7	0.1	0.2	0.5	0.2	0.3	1/3	1/3	1/3	0.2	0.5	0.3	0.1	0.7	0.2

of time penalty cost is high, the optimal route is mainly air-railway combined transportation, and more attention is paid to the length of transportation time. As the carbon emission factor of road transportation is relatively high and the transport time of railway transportation is relatively long, the carbon emission cost of Experiment 1 and 2 is relatively high and the time penalty cost is relatively high. Due to the poor safety of air transport compared with other modes of transport, the cost of compensation for damages in Experiment 4 and 5 is relatively high.

Generally speaking, in the comprehensive consideration of cost, time, energy saving, and transportation safety, whether it is air-railway combined transport or road-railway combined transport, railway occupies a large proportion, which also proves that multimodal transport based on railway transportation can indeed maximize the transport benefits.

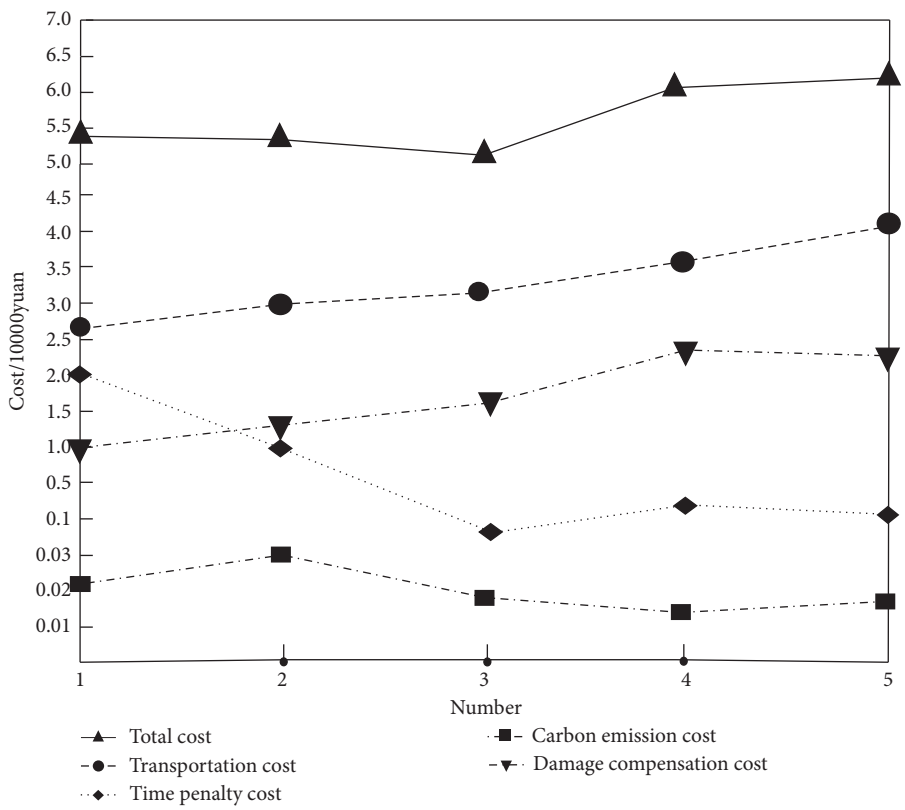


Figure 6: Sensitivity analysis of cost weights.

Table 8: Comparison of total cost of optimal path.

Optimal path	Transportation mode	Total cost (transport stability and transport safety are not considered)	Total cost (transportation stability and safety are considered)
1-3-7-9-10	Railway-highway-highway-highway	$312689	$349854

6.3. Analysis of Transportation Reliability and Transportation Safety. In order to explore the practicability of the model, all cost weights are set to 1/3, and the ant colony algorithm is used to solve the problem without considering the transportation reliability and transportation safety.

From Table 8, it can be concluded that the highway has a higher proportion in the optimal path solved without considering the transportation reliability and transportation safety. However, the transportation reliability and transportation safety of road transportation are poor, so the total cost of this optimal path in the model constructed in this paper exceeds the total cost of multimodal transportation optimal path based on railway transportation. This result is also in line with China's policy of transitioning from road transport to railway transport because railway transport is more stable and safer, and it can transport goods safely and in a timely manner. With the continuous development of other modes of transportation, in the future, China or other countries may have a multimodal transport mode based on road transport or other modes of transport. However, transportation reliability and transportation safety are the factors that must be considered in the multimodal path optimization problem; therefore, the model established in this paper is still applicable to the route selection and optimization of other transportation modes.

7. Conclusions

The paper analyzes and demonstrates the important position of railway transportation in multimodal transport and establishes a multimodal transport route selection model based on railway transportation. In view of some problems that often occur during transportation, this paper focuses on transportation reliability and transportation safety, introduces time penalty cost and damage compensation cost, and defines time delay coefficient and accident damage coefficient, respectively. Taking a multimodal transport network in China as a case, and the network is transformed according to the characteristics of the ant colony algorithm. Sensitivity analysis of cost weights proves that the multimodal transport based on railway transportation can maximize transportation efficiency. The analysis of transportation reliability and transportation safety shows that the model constructed in this paper has strong practicability and wide application range.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key R&D Program of China (2017YFB1200702), National Natural Science Foundation of China (Project nos. 52072314 and 71971182), Sichuan Science and Technology Program (Project nos. 2020YJ0268, 2020YJ0256, 2021YFQ0001, and 2021YFH0175), Chengdu Science and Technology Plan Research Program (Project nos. 2019-YF05-01493-SN, 2020-RK00-00036-ZF, and 2020-RK00-00035-ZF), Science and Technology Plan of China Railway Corporation (Project nos. P2018T001 and 2019KY10), the Natural Science Foundation of Zhejiang Province, China (LQ18G030012), and the Humanities and Social Sciences Fund of Ministry of Education, China (18YJC630190).

References

[1] Y. M. Bontekoning, C. Macharis, and J. J. Trip, "Is a new applied transportation research field emerging? A review of intermodal rail-truck freight transport literature," *Transportation Research Part A: Policy and Practice*, vol. 38, no. 1, pp. 1–34, 2003.

[2] T. Christine and L. Sabine, "Scenario-based analysis for intermodal transport in the context of service network design models," *Transportation research interdisciplinary perspectives*, vol. 2, 2019.

[3] E. Demir, W. Burgholzer, M. Hrušovský, E. Arıkan, W. Jammernegg, and T. V. Woensel, "A green intermodal service network design problem with travel time uncertainty," *Transportation Research Part B: Methodological*, vol. 93, no. 7, pp. 789–807, 2016.

[4] M. L. Chen, X. J. Zhao, X. G. Deng et al., "Multimodal transport path optimization under uncertain conditions," *Journal of Highway and Transportation Research and Development*, vol. 38, no. 1, pp. 143–150, 2021.

[5] M. Ghazanfari, L. Sara, R. Daphne, and Q. Liron, "Commuter behavior under travel time uncertainty," *Performance Evaluation*, vol. 148, 2021.

[6] M. Hrusovsky, E. Demir, W. Jammernegg, and W. Van, "Hybrid simulation and optimization approach for green intermodal transportation problem with travel time uncertainty," *International Journal of Flexible Manufacturing Systems*, vol. 30, no. 3, pp. 486–516, 2018.

[7] X. Wang and Q. Meng, "Discrete intermodal freight transportation network design with route choice behavior of intermodal operators," *Transportation Research Part B: Methodological*, vol. 95, no. 1, pp. 76–104, 2017.

[8] X. Q. Cheng and C. Jin, "Route selection problem in multimodal transportation with traffic congestion considered under low-carbon policies," *Operations Research and Management Science*, vol. 28, no. 4, pp. 67–77, 2019.

[9] S. Fazayeli, A. Eydi, and I. N. Kamalabadi, "Location-routing problem in multimodal transportation network with time windows and fuzzy demands: presenting a two-part genetic algorithm," *Computers & Industrial Engineering*, vol. 119, no. 5, pp. 233–246, 2018.

[10] H. Li and L. Su, "Multimodal transport path optimization model and algorithm considering carbon emission multitask," *The Journal of Supercomputing*, vol. 76, no. 12, pp. 9355–9373, 2020.

[11] J. Wan and S. Wei, "Multi-objective multimodal transportation path selection based on hybrid algorithm," *Journal of Tianjin University*, vol. 52, no. 3, pp. 285–292, 2019.

[12] A. Idri, M. Oukarfi, A. Boulmakoul, K. Zeitouni, and A. Masri, "A distributed approach for shortest path algorithm in dynamic multimodal transportation networks," *Transportation Research Procedia*, vol. 27, no. 12, pp. 294–300, 2017.

[13] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[14] N. Koohathongsumrit and W. Meethom, "Route selection in multimodal transportation networks: a hybrid multiple criteria decision-making approach," *Journal Of Industrial And Production Engineering*, vol. 38, no. 3, pp. 171–185, 2021.

[15] S. Liu, C. Yin, D. Chen, H. Lv, and Q. Zhang, "Cascading failure in multiple critical infrastructure interdependent networks of syncretic railway system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2021, pp. 1–14, 2021.

[16] Z. Tian, G. Sun, D. Chen, Z. Qiu, and Y. Ma, "Method for determining the valid travel route of railways based on generalised cost under the syncretic railway network," *Journal of Advanced Transportation*, vol. 2020, no. 2, 12 pages, 2020.

[17] F. Russo and U. Sansone, "The terminal cycle time in road-rail combined transport," *Wit Transactions on Ecology & the Environment*, vol. 186, pp. 875–886, 2014.

[18] A. Abderrahman, A. Ahmede, and B. Jaouad, "Robust optimization of the intermodal freight transport problem: modeling and solving with an efficient hybrid approach," *Journal of Computational Science*, vol. 30, no. 1, pp. 127–142, 2019.

[19] Y. Jiang and X. C. Zhang, "Transport plan design for rail-truck intermodal transportation," *Journal of Transporation Systems Engineering & Information Technology*, vol. 18, no. 6, pp. 222–228, 2018.

[20] X. Liu, Y. Bai, and J. Chen, "An intermodal transportation geospatial network modeling for containerized soybean shipping," *Journal of Ocean Engineering and Science*, vol. 2, no. 2, pp. 143–153, 2017.

[21] D. Chen, S. Ni, C. A. Xu, and X. Jiang, "Optimizing the draft passenger train timetable based on node importance in a railway network," *Transportation Letters*, vol. 11, no. 1, pp. 20–32, 2019.

[22] M. Verma, V. Verter, and N. Zufferey, "A bi-objective model for planning and managing rail-truck intermodal transportation of hazardous materials," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 132–149, 2011.

[23] I. Toumazis and C. Kwon, "Routing hazardous materials on time-dependent networks using conditional value-at-risk," *Transportation Research Part C: Emerging Technologies*, vol. 37, no. 3, pp. 73–92, 2013.

[24] Z. Xu, Q. Zhang, D. Chen, and Y. He, "Characterizing the connectivity of railway networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1491–1502, 2020.

[25] Y. Sun and M. Lang, "Bi-objective optimization for multimodal transportation routing planning problem based on

Pareto optimality," *Journal of Industrial Engineering and Management*, vol. 8, no. 3, pp. 1195–1217, 2015.

[26] C.-H. Chen, F.-J. Hwang, and H.-Y. Kung, "Travel time prediction system based on data clustering for waste collection vehicles," *IEICE Transactions on Information and Systems*, vol. 102, no. 7, pp. 1374–1383, 2019.

[27] Z. Elisabeth and R. V. Gloria, "Optimized allocation of straddle carriers to reduce overall delays at multimodal container terminals," *Flexible Services and Manufacturing Journal*, vol. 27, no. 2-3, pp. 300–330, 2015.

[28] S. Bret and B. Kendon, "Environmental, public health, and safety assessment of fuel pipelines and other freight transportation modes," *Applied Energy*, vol. 171, pp. 266–276, 2016.

[29] T. Yang, X. Peng, D. Chen, F. Yang, and M. Muneeb Abid, "Research on trans-region integrated traffic emergency dispatching technology based on multi-agent," *Journal of Intelligent & Fuzzy Systems*, vol. 385, no. 5, pp. 5763–5774, 2020.

[30] H. Y. Luo, J. Yang, and X. Nan, "Path and transport mode selection in multimodal transportation with time window," in *Proceedings of the 3rd IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 162–166, Chongqing, China, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6817013, 13 pages
https://doi.org/10.1155/2021/6817013

Research Article

Two-Sided Matching on Comprehensive Transportation Network Emergency Vehicles' Allocation

Kunwei Xie,[1] **Heying Xu**,[1] **and Hongxia Lv**[1,2,3]

[1]*School of Transportation and Logistics, Southwest Jiaotong University, Chengdu 610031, China*
[2]*National and Local Joint Engineering Laboratory of Comprehensive Intelligent Transportation, Southwest Jiaotong University, Chengdu 610031, China*
[3]*National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Chengdu 610031, China*

Correspondence should be addressed to Heying Xu; 380013157@qq.com

Received 5 April 2021; Revised 27 May 2021; Accepted 17 June 2021; Published 5 July 2021

Academic Editor: Chi-Hua Chen

In emergency rescue, the allocation of comprehensive transportation network emergency vehicles often affects the efficiency of the whole rescue process. In the context of disasters, this paper researches the one-to-many two-sided matching problem between the emergency vehicles and the materials to be transported. Firstly, based on the needs of both parties involved in the matching, the satisfaction evaluation systems are constructed; with the goal of maximizing the weighted satisfaction of the affected areas and vehicles, the optimization model of the materials and emergency vehicles matching is established; then, an improved National Intern Matching Program (NIMP) algorithm is designed to solve the model, which is based on the k: 1 experimental pairing and updating ideas, and can take into account the capacity and destination constraints of vehicles in the matching process. Finally, through the calculation of an example, the matching scheme can make the satisfaction of material transportation reach 0.7392, and the simulation analysis proves that the scheme keeps certain stability in risky conditions.

1. Introduction

Emergency causes huge losses and social impact because of the suddenness, unpredictability, and diffusion. In the process of emergency response, transportation network often plays an important role in ensuring the transportation of materials and transfer of disaster victims. A scientific and reasonable allocation scheme of emergency vehicles can make limited transportation resources fit the transportation need between the supply point and the affected area to the maximum extent, which is of great significance for the improvement of emergency capacity of the whole society. However, disasters cause different degrees of damage to the road, resulting in one or more transportation networks getting paralyzed, which is not conducive to the emergency work.

With the rapid construction of transportation infrastructure and modern comprehensive transportation system, the division of labor among various modes of transportation becomes clearer and closer. Therefore, the formulation of emergency transportation schemes should be based on different needs of affected areas, different characteristics of demand materials, and different modes of transportation, to realize the reasonable and efficient utilization of the emergency vehicles. Comprehensive transportation network emergency vehicles refer to all kinds of comprehensive transportation mobile equipment that can be used in the territory and the whole rescue scope in the process of joint prevention and control, emergency rescue, disaster victims transfer, post disaster recovery, and other work in order to control the disaster and reduce losses under the constraints of certain fixed facilities and channel transport capacity.

Existing research on material transportation optimization mainly includes the emergency vehicle route planning, distribution, and scheduling.

On the emergency vehicle route planning, Zhang et al. [1] established the shortest path model considering the road

conditions and the shortage of demand resources. Hu [2] used a linear programming model to plan the route. Özdamar et al. [3] built a combined transportation path model based on node clustering.

On the vehicle distribution, Rawls et al. [4] proposed a two-stage mixed integer programming location model. Bozorgi-Amiri et al. [5] established an emergency material distribution model considering the uncertainty of demand location.

On the vehicle scheduling, Afshar et al. [6], Sabouhi et al. [7], and Huang et al. [8] researched the emergency vehicle scheduling model from the perspective of logistics operation. Laporte et al. [9], Arda et al. [10], Duan, et al. [11], and Klibi et al. [12] proposed the multistage model to optimize vehicle scheduling.

The above papers mainly solve the problems of emergency vehicle distribution and scheduling, and the limitations are as follows. (1) Existing literature considers the same model of vehicles, and materials are homogeneous. (2) Existing literature usually ignores train, and flight operation is limited by timetable. Actually, in emergency transportation, different vehicles have different loading attributes and some of them may have fixed operating period. These constraints make existing research neither clarify the matching relationship between materials and vehicles nor put into use directly.

Considering that the two-sided matching theory can be used to solve the matching problem between two objects and the development of emergency management information platform makes it easier to bring the information of various rescue resources into the unified platform for centralized management, it is possible for us to research how to find a satisfactory matching between materials and vehicles with this theory.

Two-sided matching theory was first proposed by Gale and Shapley [13], and Roth [14, 15] clearly defined the concepts of "two-sided" and "two-sided matching." "Two-sided" refers to the participants in the market who belong to two disjoint sets, and "matching" refers to the two-sided nature of market exchange. Since both parties have the preference to get stable matching, two-sided matching theory is used to study the matching process of disjoint parties with stable preference.

The process of emergency vehicles' allocation is participated by the vehicles and materials that need to be transported, and the core is to find an optimal resource allocation scheme in the context of emergency so that both sides can find the matching objects which meet their own needs. Therefore, combined with the characteristics of strong timeliness, high uncertainty, and weak economy of emergency rescue, this paper proposes an emergency vehicle allocation method based on the two-sided matching theory, which has three main contributions:

(1) The application of two-sided matching theory is extended

(2) A two-sided matching model considering materials' loading demands is established, which can guide the matching of materials with specific emergency vehicles

(3) An example validates the stability and operability of emergency vehicles' allocation scheme under certain risks

The remainder of the paper is organized as follows. Section 2 introduces the problem and presents an analysis on matching satisfaction. Section 3 formulates a two-side matching model, and an improved Nation Intern Matching Program (NIMP) algorithm is proposed in Section 4. Thereafter, a matching example is conducted in Section 5. Section 6 concludes the paper and provides potential future work.

2. Matching Satisfaction Calculation

2.1. Problem Presentation. Suppose that there are disaster affected areas and each area needs M materials. We use R to denote the set of materials to transport:

$$R = \{R_1, R_2, \ldots, R_i, \ldots, R_I\}^T, \\ = \begin{bmatrix} r_{11} & \cdots & r_{1M} \\ \vdots & \ddots & \vdots \\ r_{I1} & \cdots & r_{IM} \end{bmatrix}, \tag{1}$$

where R_i denotes the set of materials demand of area i, $R_i = \{r_{i1}, \ldots, r_{im}, \ldots, r_{iM}\}$, and r_{im} denotes the material m required for area i, $i = 1, 2, \ldots, I$ and $m = 1, 2, \ldots M$.

Emergency vehicle set V is composed of J vehicles, $V = \{V_1, \ldots, V_j, \ldots, V_J\}$. V_j denotes vehicle j, $j = 1, 2, \ldots, J$.

Mapping $\mu\ R \cup S \longrightarrow R \cup S$ denotes the matching result between r_{im} and V_j. If the rim is transported by V_j, $r_{im} \in \mu(V_j)$ and $V_j \in \mu(r_{im})$.

2.2. Model Assumptions. Before modeling, we make the following assumptions: (1) the materials needed for one disaster affected area can be transported by different vehicles; (2) each vehicle is only used to meet the transportation demand of one disaster area, but it can transport different materials needed in this area; (3) each material rim cannot be separated or transshipped during transportation; (4) if vehicle V_j is also the required material for affected areas, we assume a virtual good r_{im} in set R, and this r_{im} has the following characteristics: $r_{im} = V_j$; the weight of r_{im} is 0; r_{im} can only be transported by V_j; (5) the number and capacity of available vehicles are limited.

Because decision makers can obtain the information of the affected areas' material demand and comprehensive transportation network emergency vehicles through the emergency management information platform, there is a two-sided matching market between the demand materials and emergency vehicles based on the information platform. According to the model assumptions, this is a one-to-many two-sided matching market. One affected area in this market can be served by multiple vehicles, which forms a one-to-

many matching relationship between one area and vehicles; at the same time, one vehicle can transport a variety of materials, so that the matching market between vehicles and materials can be regarded as one-to-many matching relationships restricted by the materials' destination. The operating mechanism of this matching market is shown in Figure 1.

2.3. Matching Satisfaction Index Systems. The premise of establishing the demand materials and emergency transportation vehicles' matching optimization model is the construction of the evaluation index systems on both sides so that we can evaluate the satisfaction of different matching schemes. The parameters related to evaluation models are given in Table 1.

We use $U = \{u1, u2, \ldots, u4\}$ to denote the satisfaction evaluation index set of demand materials and $S = \{s1, s2, \ldots, s4\}$ to denote the satisfaction evaluation index set of comprehensive transportation network emergency vehicles. Combined with the research on the evaluation of emergency material allocation [16–20] and vehicle-material matching problem [21–25] and considering the potential applications of the proposed method in high-speed rail systems [26, 27], the evaluation index systems on both sides are established.

For each material to be transported, the optimal scheme is to make the transportation time as short as possible. And, the transportation route should be reliable, so as to avoid the routes damage making the material transportation interruption. Therefore, on the basis of the loading characteristics of materials, we establish the material satisfaction evaluation index system as follows.

(1) Arrival time index $u_1^{im,j}$: in emergency transportation, higher satisfaction of materials means earlier arrival time transport by vehicles:

$$u_1^{im,j} = \frac{T_{im}^{\min}}{T_{im}^{j}}. \tag{2}$$

(2) Delay probability index $u_2^{im,j}$: the delay probability of V_j reflects the reliability of vehicle route. Higher material satisfaction requires higher reliability and lower delay probability:

$$u_2^{im,j} = \frac{\xi_j^{\min}}{\xi_j}. \tag{3}$$

(3) Loading condition index $u_3^{im,j}$: different vehicles have different loading conditions, such as thermal, container, and flat, and materials evaluate different vehicles according to their own requirements:

$$u_3^{im,j} \in [0, 1]. \tag{4}$$

(4) Cargo capacity index $u_4^{im,j}$: the weight of materials should not exceed the vehicles' weight limitations:

$$u_4^{im,j} = \begin{cases} 0, & g_{im} > c_j, \\ 1, & g_{im} \le c_j. \end{cases} \tag{5}$$

The satisfaction value of the materials in relation to vehicles is calculated as follows. In equation (6), ω_p^{im} denotes the weight of $u_p^{im,j}$, $\sum_{p=1}^{4} \omega_p^{im} = 1$, and the satisfaction value matrix is R_V, $R_V = (\lambda_{jim})_{J\times I\times M}$:

$$\lambda_{jim} = \prod_{p=1}^{4} \left(u_p^{im,j}\right)^{\omega_p^{im}}, \quad \forall j. \tag{6}$$

The rescue mission of each vehicle is to deliver materials to the respective destinations reliably and effectively. In actual rescue, emergencies usually attack routes thus prolonging the transportation time. So, we introduce control coefficient κ of arrival time, in response to the risk that materials cannot be delivered on schedule. Combined with vehicle loading operation, we establish the vehicle satisfaction evaluation index system as follows.

(1) Satisfaction time window index $s_1^{j,im}$: the arrival time of emergency transportation is usually uncertain. Therefore, we reserve satisfactory time window and keep a certain time margin to ensure the stability of the scheme:

$$s_1^{j,im} = \begin{cases} 1, & T_{im}^{j} + \kappa \cdot \widehat{T_{im}^{j}} \le TW_{im}^{\min}, \\ 1 - 2\left(\dfrac{T_{im}^{j} + \kappa \cdot \widehat{T_{im}^{j}} - TW_{im}^{\min}}{TW_{im}^{\max} - TW_{im}^{\min}}\right)^2, & TW_{im}^{\min} < T_{im}^{j} + \kappa \cdot \widehat{T_{im}^{j}} \le \dfrac{TW_{im}^{\min} + TW_{im}^{\max}}{2}, \\ 2\left(\dfrac{T_{im}^{j} + \kappa \cdot \widehat{T_{im}^{j}} - TW_{im}^{\max}}{TW_{im}^{\max} - TW_{im}^{\min}}\right)^2, & \dfrac{TW_{im}^{\min} + TW_{im}^{\max}}{2} < T_{im}^{j} + \kappa \cdot \widehat{T_{im}^{j}} \le TW_{im}^{\max}, \\ 0, & T_{im}^{j} + \kappa \cdot \widehat{T_{im}^{j}} > TW_{im}^{\max}. \end{cases} \tag{7}$$

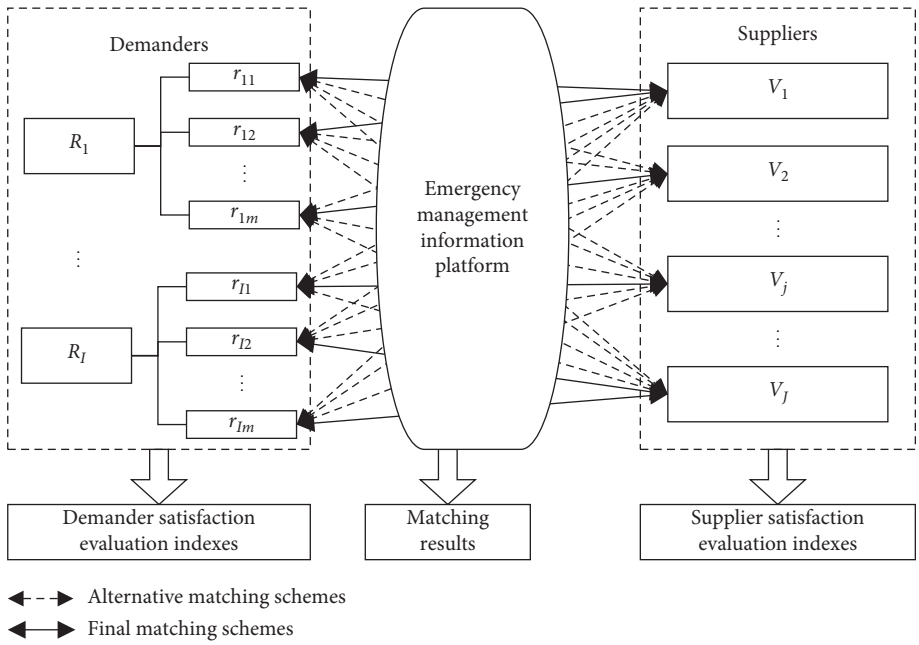


Figure 1: Two-sided matching mode of emergency platform.

Table 1: Notation statement.

Set	Statement
U	Set of demand materials satisfaction evaluation index
S	Set of emergency vehicles' satisfaction evaluation index
lo_j	The loadable materials set of vehicle V_j
Parameter	Statement
T_{im}^j	Estimated arrival time of rim transported by V_j
$T_{im}^{\min}$	Minimum value of T_{im}^j for each r_{im}
$\overline{T}_{im}^j$	Maximum positive deviation of T_{im}^j affected by emergency
$[TW_{im}^{\min}, TW_{im}^{\max}]$	Satisfaction arrival time window of r_{im}
ξ_j	Delay probability of V_j
$\xi_j^{\min}$	Minimum value of ξ_j
d_{im}^j	Deadhead kilometers of r_{im} transported by V_j
$d_j^{\min}, d_j^{\max}$	Minimum and maximum value of d_{im}^j for each V_j
t_{im}^j	Handling time of r_{im} transported by V_j
$t_j^{\min}, t_j^{\max}$	Minimum and maximum value of t_{im}^j for each V_j
g_{im}	Estimated weight of r_{im}
c_j	Cargo capacity of V_j
κ	Control coefficient of arrival time

(2) Empty driving index $s_2^{j,im}$: the origin of V_j should be the same as the rim to avoid empty driving:

$$s_2^{j,im} = \frac{d_j^{\max} - d_{im}^j}{d_j^{\max} - d_j^{\min}} + \varepsilon. \tag{8}$$

In order to ensure materials are transported as many as possible, we use minimum ε to make $s_2^{j,im} > 0$.

(3) Handling time index $s_3^{j,im}$: the handling time reflects the complexity of handling operations:

$$s_3^{j,im} = \begin{cases} 0, & r_{im} \notin lo_j, \\ \dfrac{t_j^{\max} - t_{im}^j}{t_j^{\max} - t_j^{\min}} + \varepsilon, & r_{im} \in lo_j. \end{cases} \tag{9}$$

In addition, in order to ensure materials are transported as much as possible, we use minimum ε to make $s_2^{j,im} = 0$ if and only if $r_{im} \notin lo_j$.

(4) Cargo capacity index $s_4^{j,im}$: the cargo capacity of vehicles should exceed the materials' weight:

$$s_4^{j,im} = \begin{cases} 0, & g_{im} > c_j, \\ 1, & g_{im} \le c_j. \end{cases} \tag{10}$$

The satisfaction value of the vehicles in relation to materials is calculated as follows. In equation (11), ω_q^j denotes the weight of $s_q^{j,im}$, $\sum_{q=1}^{4} \omega_q^j = 1$, and the satisfaction value matrix is V_R, $V_R = (\theta_{jim})_{J\times I\times M}$:

$$\theta_{jim} = \prod_{p=1}^{4} \left(s_p^{j,im}\right)^{\omega_q^j}, \quad \forall i, m. \tag{11}$$

3. Two-Sided Matching Model

Since different disaster affected areas have different urgency and materials play various roles in different areas, α_{im} denoting demand urgency weight of rim is used to make the matching scheme satisfy the total material demand to the maximum extent. We aim at optimal satisfaction on both sides and establish a one-to-many two-sided matching model:

$$\max F_1 = \sum_{i=1}^{I} \sum_{m=1}^{M} \alpha_{im} \cdot \sum_{j=1}^{J} \lambda_{jim} \cdot x_{jim}, \tag{12}$$

$$\max F_2 = \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{j=1}^{J} \theta_{jim} \cdot x_{jim}, \tag{13}$$

$$s.t. \quad \sum_{j=1}^{J} x_{jim} \le 1, \quad (\forall i, m), \tag{14}$$

$$\left(\sum_{m=1}^{M} x_{jim}\right) \cdot \left(\sum_{m=1}^{M} x_{ji'm}\right) = 0, \quad (\forall j, i, i'; i \ne i'), \tag{15}$$

$$\sum_{i=1}^{I} \sum_{m=1}^{M} g_{im} \cdot x_{jim} \le c_j, \quad (\forall j), \tag{16}$$

$$\left(\lambda_{jim} - \lambda'\right) \cdot x_{jim} \ge 0, \quad (\forall j, i, m), \tag{17}$$

$$\left(\theta_{jim} - \theta'\right) \cdot x_{jim} \ge 0, \quad (\forall j, i, m), \tag{18}$$

$$x_{jim} = 1, \quad \left(\forall j, i; r_{im} = S_j\right), \tag{19}$$

$$x_{jim} \in \{0, 1\}, \quad (\forall j, i, m). \tag{20}$$

Equations (12) and (13) are the object of the optimization model. Equation (12) means maximum satisfaction of affected areas. Equation (13) represents the optimal object of vehicles. Equation (14) limits one material matches at most one vehicle. Equation (15) indicates that one vehicle is only used to meet the transportation demand of one disaster area. Equation (16) gives the cargo capacity constraint, in which the cargo capacity of the vehicle should exceed the weight of matching materials. Equations (17) and (18) are the minimum matching satisfaction constraint: if and only if r_{im} and $\theta_{jim} > \theta\prime$, matching between r_{im} and V_j is possible. Equation (19) denotes if $r_{im} = V_j$, $x_{jim} = 1$. Equation (20) denotes the range of decision variable x_{jim}. $x_{jim} = 1$ denotes r_{im} matches V_j. $x_{jim} = 0$ denotes r_{im} does not match V_j.

4. Algorithm Design

National Intern Matching Program (NIMP) algorithm is an efficient algorithm which has been applied in one-to-many matching problem and successfully served many markets. The NIMP is consisted of a matching phase and a tentative assignment and update phase. In the kth step of the matching phase (which is called k: 1 step), the demander-supplier pairs such that the demander is top ranked on the supplier's ranking and the supplier is kth ranked by the demander are sought to find (here we assume one supplier serves multiple demanders). If such matches are found, the algorithm proceeds to tentatively assign and update [28]. It is proved that NIMP algorithm can generate stable matching in given preference and obtain the optimal stable matching of one party under strict preference sequence.

Because there are two matching markets in the model: one vehicle matches multiple materials and one affected area matches multiple vehicles, and NIMP algorithm is used to solve only one matching market problem. This paper proposes an improved NIMP algorithm to solve the model. The definition of algorithm parameters is as follows.

n denotes iteration variable; $R_1^{(n)}$ denotes unmatched set of materials in the nth iteration, $R_2^{(n)}$ denotes matched set of materials, $R_1^{(n)} \cap R_2^{(n)} = \varnothing$, $R_1^{(n)} \cup R_2^{(n)} = R$; $P_{V_j}^{(n)}$ is the set that V_j is nth ranked by $r_{im} \in P_{V_j}^{(n)}$, and $\lambda'_{jim} > \lambda'$, $\theta_{jim} > \theta'$; $P_{V_j,i}^{(n)} \subseteq P_{V_j}^{(n)}$, each $r_{im} \in P_{V_j,i}^{(n)}$ meets $\sum_{r_{im} \in P_{V_j,i}^{(n)}} \lambda'_{jim} = \max \sum_{r_{im} \in (P_{V_j}^{(n)} \cap R_i)} \lambda'_{jim}$, $\sum_{r_{im} \in P_{V_j,i}^{(n)}} g_{im} \le c_j$, and belongs to the same R_i; $\mu^{(n)}$ denotes the matching scheme in the nth iteration; $c_j^{(n)}$ denotes the remaining cargo capacity of V_j before nth iteration.

Step 1: calculate the satisfaction matrix $V_R = (\theta_{jim})_{J\times I\times M}$ and $R'_V = (\lambda'_J \times I \times M) = ((\lambda_{jim})_J \cdot \alpha_{im}) I \times M$, and obtain the preference order lists submitted by the demand materials and vehicles.

Step 2: initialize variables $n = 1$. Determine whether $r_{im} = V_j$ is present; if yes, $\mu^{(1)}(r_{im}) = V_j$, $\mu^{(1)}(V_j) = r_{im}$, $R_1^{(1)} = R - \{r_{im}\}$, and $R_2^{(1)} = R_2^{(1)} \cup \{r_{im}\}$, and turn to step 3; else, $R_1^{(1)} = R$, $R_2^{(1)} = \varnothing$, $P_{V_j}^{(1)} = \varnothing$, $\mu^{(1)}(r_{im}) = \varnothing$, $\mu^{(1)}(V_j) = \varnothing$, and turn to step 3.

Step 3: find the nth choice V_j of each r_{im}. Identify the set $P_{V_j}^{(n)}$ of V_j, $P_{V_j}^{(n)} \subseteq R_1^{(n)}$. If one material has the

same satisfaction for different vehicles, take a random method to determine a virtual strict preference order. Match V_j and r_{im} as the following steps.

(1) $j = 1$.
(2) Determine whether $P_{V_j}^{(n)} = \varnothing$ is true; if yes, turn to step 3.8; else, turn to step (3).
(3) Determine whether $\mu^{(n)}(V_j) = \varnothing$ is true; if yes, turn to step (4); else, turn to step (3).
(a) Determine whether $|P_{V_j}^{(n)}| = 1$ is true; if yes, let $\mu^{(n+1)}(r_{im}) = V_j$, $\mu^{(n+1)}(V_j) = r_{im}$, $R_1^{(n+1)} = R_1^{(n)} - P_{V_j}^{(n)}$, $R_2^{(n+1)} = R_2^{(n)} \cup P_{V_j}^{(n)}$, and $c_j^{(n+1)} = c_j^{(n)} - g_{im}$, and turn to step (8); else, turn to step (b).
(b) Determine whether there are different material need to be transported in the same direction in set $P_{V_j}^{(n)}$; if yes, turn to step (c); else, only if $\lambda'_{jim} = \max\{\lambda'_{jim}\}$, let $\mu^{(n+1)}(r_{im}) = V_j$, $\mu^{(n+1)}(V_j) = r_{im}$, $R_1^{(n+1)} = R_1^{(n)} - P_{V_j}^{(n)}$, $R_2^{(n+1)} = R_2^{(n)} \cup P_{V_j}^{(n)}$, and $c_j^{(n+1)} = c_j^{(n)} - g_{im}$, and turn to step (8).
(d) Let $\mu^{(n+1)}(V_j) = P_{V_j,i}^{(n)}$, $\mu^{(n+1)}(r_{im}) = V_j$ $(r_{im} \in P_{V_j,i}^{(n)})$, $R_1^{(n+1)} = R_1^{(n)} - P_{V_j,i}^{(n)}$, $R_2^{(n+1)} = R_2^{(n)} \cup P_{V_j,i}^{(n)}$ $c_j^{(n+1)} = c_j^{(n)} - \sum_m g_{im}$, and turn to step (8).
(4) Find the set $P_{V_j,i}^{(n)}$; determine whether $P_{V_j,i}^{(n)}$ is empty; if yes, turn to step (8); else turn to step (5).
(5) Determine whether $P_{V_j,i}^{(n)}$ and $\mu^{(n)}(V_j)$ belong to the same R_i, if not, turn to step (6); if yes, let $\mu^{(n+1)}(V_j) = \mu^{(n)}(V_j) \cup P_{V_j,i}^{(n)}$, $\mu^{(n+1)}(r_{im}) = V_j$ $(r_{im} \in P_{V_j,i}^{(n)})$, $R_1^{(n+1)} = R_1^{(n)} - P_{V_j,i}^{(n)}$, $R_2^{(n+1)} = R_2^{(n)} \cup P_{V_j,i}^{(n)}$, $c_j^{(n+1)} = c_j^{(n)} - \sum_m g_{im}$, and turn to step (8).
(6) Determine whether $\sum_{r_{im} \in P_{V_j,i}^{(n)}} \lambda'_{jim} > \sum_{r_{im} \in \mu^{(n)}(V_j)} \lambda'_{jim}$ is true; if yes, turn to step (7); if not, determine whether there is $r_{im} \in P_{V_j}^{(n)}$ with the same destination as $r_{im} \in \mu^{(n)}(V_j)$; if yes, let $\mu^{(n+1)}(V_j) = \mu^{(n)}(V_j) \cup P_{V_j,\mu^{(n)}(V_j)}^{(n)}$, $\mu^{(n+1)}(r_{im}) = V_j(r_{im} \in P_{V_j,i}^{(n)})$, $R_1^{(n+1)} = R_1^{(n)} - P_{V_j,\mu^{(n)}(V_j)}^{(n)}$, $R_2^{(n+1)} = R_2^{(n)} \cup P_{V_j,\mu^{(n)}(V_j)}^{(n)}$ $c_j^{(n+1)} = c_j^{(n)} - \sum_m g_{im}$, turn to step (8); else, turn to step (8).
(7) Let $R_1^{(n+1)} = R_1^{(n)} \cup \mu^{(n)}(V_j)$, $R_2^{(n+1)} = R_2^{(n)} - \mu^{(n)}(V_j)$, $P_{V_{j'}}^{(n)} = P_{V_{j'}}^{(n)} \cup \mu^{(n)}(V_j)$ (V_j is the nth choice of r_{im}), recalculate $\mu^{(n)}(V_j)$; let $\mu^{(n+1)}(r_{im}) = V_j(r_{im} \in P_{V_j,i}^{(n)})$, $R_1^{(n+1)} = R_1^{(n+1)} - P_{V_j,i}^{(n)}$, $R_2^{(n+1)} = R_2^{(n+1)} \cup P_{V_j,i}^{(n)}$, and $c_j^{(n+1)} = c_j^{(1)} - \sum_m g_{im}$, and turn to step (8).
(8) Determine whether $j \geq J$ is true; if yes, turn to step 4; else, $j = j + 1$ and turn to step (2).

Step 4: determine whether $R_1^{(n)} \neq \varnothing$ is true; if yes, end the loop and turn to step 5; if not, determine whether $n \geq J$ is true; if not, $n = n + 1$ and repeat steps 3-4; else, end the loop and turn to step 5.

Step 5: determine whether there are $\mu^{(n+1)}(V_{j1})$, $\mu^{(n+1)}(V_{j2})$, and $\mu^{(n+1)}(V_{j3})$ which satisfy that

① $\mu^{(n+1)}(V_{j1})$ and $\mu^{(n+1)}(V_{j2})$ belong to the same R_i different from $\mu^{(n+1)}(V_{j3})$.
② $c_{j1}^{(n+1)} \geq \sum_{r_{im} \in \mu^{(n+1)}(V_{j2})} g_{im}$.
③ $\sum_{r_{im} \in \mu^{(n+1)}(V_{j3})} \lambda'_{j2im} - \sum_{r_{im} \in \mu^{(n+1)}(V_{j3})} \lambda'_{j3im} > \sum_{r_{im} \in \mu^{(n+1)}(V_{j2})} \lambda'_{j2im} - \sum_{r_{im} \in \mu^{(n+1)}(V_{j2})} \lambda_{j1im}$. If yes, $\mu^{(n+2)}(r_{im}) = V_{j1}(r_{im} \in \mu^{(n+1)}(V_{j2}))$, $\mu^{(n+2)}(V_{j1}) = \mu^{(n+1)}(V_{j1}) \bigcup \mu^{(n+1)}(V_{j2})$, $\mu^{(n+2)}(V_{j2}) = \mu^{(n+1)}(V_{j3})$, $\mu^{(n+2)}(r_{im}) = V_{j2}(r_{im} \in \mu^{(n+1)}(V_{j3}))$, $\mu^{(n+2)}(V_{j3}) = \varnothing$, and output, else, output.

5. Simulation Results and Analysis

To verify the validity of the two-sided matching model, we use the data of 4 affected locations (R_1, R_2, R_3, R_4), and each location needs 4 materials $(r_{11}, r_{12}, \ldots, r_{44})$. At the same time, we collect information about 12 vehicles $(V_1, V_2, \ldots, V_{12})$ used for emergency transportation. Information on materials and vehicles is shown in Tables 2 and 3.

In addition, we calculate the estimated arrival time as follows:

$$T_{im}^{j} = \begin{bmatrix} 21.8 & 21.89 & 23.79 & 21.71 & 22 & 21.79 & 26.16 & 21.99 & 35.93 & 35.46 & 42.14 & +\infty & 34.54 & 35.26 & 40.51 & 35.09 \\ 21.4 & 22.46 & 23.92 & 21.91 & 22 & 22.32 & 26.18 & 22.29 & 35.58 & 36.02 & 41.92 & +\infty & 34.74 & 35.81 & 40.41 & 35.34 \\ 23.8 & 23.87 & 26.64 & 23.66 & 27.05 & 26.77 & 31.84 & 27.14 & 17.64 & 17.40 & 24.64 & +\infty & 31.43 & 32.17 & 37.94 & 31.9 \\ 23.4 & 24.24 & 26.29 & 23.86 & 27.05 & 27.21 & 31.26 & 27.44 & 17.29 & 17.80 & 23.52 & 16.41 & 31.63 & 32.58 & 37.09 & 32.15 \\ 28.41 & 28.38 & 28.60 & 28.56 & 28.78 & 28.57 & 30.5 & 28.9 & 46.66 & 46.07 & 49.31 & 47.13 & 35.48 & 36.00 & 38.32 & 36.13 \\ 30.77 & 30.38 & 30.79 & 30.68 & 30.98 & 30.57 & 32.74 & 30.98 & 49 & 48.07 & 51.67 & 49.13 & 37.6 & 38.00 & 40.62 & 38.23 \\ 30.91 & 31.27 & 32.07 & 31.24 & 35.67 & 35.58 & 37.74 & 36.03 & 21.99 & 22.08 & 25.01 & 22.93 & 41.36 & 42.07 & 44.18 & 41.93 \\ 33.73 & 34.29 & 35.16 & 34.18 & 38.57 & 38.58 & 40.86 & 38.99 & 24.82 & 25.10 & 28.19 & 25.93 & 44.3 & 45.08 & 47.33 & 44.88 \\ 9.15 & 9.50 & 11.97 & 8.8 & 9.20 & 9.06 & 14.48 & 9.05 & 15.13 & 14.98 & 22.88 & 15.54 & 10.72 & 11.74 & 18.15 & 11.1 \\ 6.17 & 6.51 & 9.23 & 5.8 & 7.22 & 7.06 & 12.63 & 7.11 & 7.03 & 6.98 & 14.92 & 7.48 & 7.72 & 8.76 & 15.22 & 8.07 \\ 10.15 & 10.48 & 13.25 & 9.8 & 8.20 & 8.05 & 13.84 & 8.05 & 11.13 & 10.97 & 19.42 & 11.54 & 5.72 & 6.73 & 13.6 & 6.1 \\ 10.17 & 10.50 & 13.14 & 9.8 & 7.22 & 7.05 & 12.51 & 7.11 & 9.03 & 8.97 & 16.74 & 9.48 & 5.72 & 6.74 & 13.07 & 6.07 \end{bmatrix}. \quad (21)$$

Table 2: Information on materials.

Materials	Weight (t)	Time window (h) $[TW_{im}^{\min}, TW_{im}^{\max}]$	α_{im}	ω_p^{im}			
				ω_1^{im}	ω_2^{im}	ω_3^{im}	ω_4^{im}
r_{11}	36	[6, 24]	0.137	0.5	0.3	0.05	0.15
r_{12}	188	[24, 48]	0.098	0.3	0.3	0.2	0.2
r_{13}	940	[48, 72]	0.059	0.3	0.2	0.2	0.3
r_{14}	16	[12,48]	0.019	0.3	0.2	0.2	0.3
r_{15}	20	[6, 24]	0.027	0.5	0.3	0.05	0.15
r_{16}	80	[24, 48]	0.019	0.3	0.3	0.2	0.2
r_{17}	1200	[48, 72]	0.004	0.3	0.2	0.2	0.3
r_{24}	8	[12, 52]	0.012	0.4	0.2	0.1	0.3
r_{31}	34	[6, 24]	0.191	0.5	0.3	0.05	0.15
r_{32}	150	[24, 48]	0.082	0.3	0.3	0.2	0.2
r_{33}	1600	[48, 72]	0.027	0.3	0.2	0.2	0.3
r_{34}	0 $(r_{34} = V_4)$	[12, 36]	0.137	0.5	0.4	0.05	0.05
r_{41}	22	[6, 24]	0.082	0.5	0.3	0.05	0.15
r_{42}	130	[24, 48]	0.035	0.3	0.3	0.2	0.2
r_{43}	1500	[48, 72]	0.012	0.3	0.2	0.2	0.3
r_{44}	10	[12, 36]	0.059	0.4	0.3	0.2	0.1

Table 3: Information on vehicles.

Vehicles	Cargo capacity (t)	Loadable materials set	ξ_j	Vehicle type	ω_q^j			
					ω_1^j	ω_2^j	ω_3^j	ω_4^j
V_1	20	R-$\{r_{34}\}$, $\{r_{34}\}$	0.6	Truck	0.3	0.2	0.25	0.25
V_2	20	R-$\{r_{34}\}$	0.6	Truck	0.3	0.2	0.25	0.25
V_3	20	R-$\{r_{34}\}$	0.5	Truck	0.3	0.2	0.25	0.25
V_4	2	$\{r_{11}, r_{21}, r_{31}, r_{34}, r_{41}\}$	0.5	Ambulance	0.7	0.1	0.1	0.1
V_5	2000	R	0.2	Freight train	0.2	0.4	0.2	0.2
V_6	2000	R	0.2	Freight train	0.2	0.4	0.2	0.2
V_7	2000	R	0.22	Freight train	0.2	0.4	0.2	0.2
V_8	2000	R	0.22	Freight train	0.2	0.4	0.2	0.2
V_9	50	R	0.1	Helicopter	0.3	0.4	0.15	0.15
V_{10}	50	R	0.1	Helicopter	0.3	0.4	0.15	0.15
V_{11}	50	R	0.1	Helicopter	0.3	0.4	0.15	0.15
V_{12}	50	R	0.1	Helicopter	0.3	0.4	0.15	0.15

The maximum positive deviation of T_{im}^j is as follows:

$$\widehat{T}_{im}^j = \begin{bmatrix} 20 & 20.06 & 19.75 & 20.03 & 21 & 21.01 & 21 & 21.03 & 34.23 & 34 & 34.4 & 34.3 & 33.94 & 34 & 34.06 & 34.19 \\ 20.5 & 20.56 & 20.25 & 20.53 & 21.5 & 21.52 & 21.5 & 21.53 & 34.73 & 34.5 & 34.9 & 34.8 & 34.44 & 34.5 & 34.56 & 34.69 \\ 33 & 33.09 & 33.19 & 32.96 & 39.08 & 39 & 39.11 & 39.26 & 23.91 & 23.94 & 24 & 13.9 & 46.24 & 46.39 & 46.11 & 46.5 \\ 33.75 & 33.84 & 33.94 & 33.71 & 39.83 & 39.75 & 39.86 & 40.01 & 24.66 & 24.69 & 24.75 & 24.62 & 46.99 & 47.14 & 46.86 & 47.25 \\ 13.49 & 13.53 & 13.32 & 13.5 & 13.99 & 14 & 13.00 & 14.01 & 22.65 & 22.5 & 22.77 & 22.57 & 17.5 & 17.54 & 17.59 & 17.67 \\ 14.49 & 14.53 & 14.32 & 14.5 & 14.99 & 15 & 14.99 & 15.01 & 23.65 & 23.5 & 23.77 & 23.57 & 18.5 & 18.54 & 18.59 & 18.67 \\ 14.92 & 14.94 & 15 & 14.9 & 17.54 & 17.5 & 17.55 & 17.62 & 10.49 & 10.5 & 10.53 & 10.47 & 20.5 & 20.57 & 20.44 & 20.62 \\ 16.42 & 16.46 & 16.5 & 16.4 & 19.04 & 19 & 19.05 & 19.12 & 11.99 & 12 & 12.03 & 11.97 & 22 & 22.07 & 21.94 & 22.12 \\ 2.10 & 2.10 & 2.07 & 2.1 & 2.40 & 2.4 & 2.40 & 2.40 & 3.93 & 3.9 & 3.95 & 3.91 & 3 & 3.01 & 3.02 & 3.03 \\ 1.20 & 1.21 & 1.22 & 1.2 & 1.80 & 1.8 & 1.81 & 1.82 & 1.50 & 1.5 & 1.50 & 1.49 & 2.1 & 2.11 & 2.09 & 2.82 \\ 2.40 & 2.40 & 2.37 & 2.4 & 2.10 & 2.1 & 2.10 & 2.10 & 2.73 & 2.7 & 2.75 & 2.71 & 1.5 & 1.51 & 1.52 & 1.53 \\ 2.40 & 2.41 & 2.42 & 2.4 & 1.80 & 1.8 & 1.81 & 1.82 & 2.10 & 2.1 & 2.10 & 2.09 & 1.5 & 1.51 & 1.49 & 1.52 \end{bmatrix}. \tag{22}$$

The distances d_{im}^{j} between r_{im} and V_j are as follows:

$$d_{im}^{j}=\begin{bmatrix}
40 & 45 & 20 & 42 & 21 & 22 & 21 & 23 & 130 & 112 & 144 & 120 & 70 & 75 & 80 & 90\\
40 & 45 & 20 & 42 & 21 & 22 & 21 & 23 & 130 & 112 & 144 & 120 & 70 & 75 & 80 & 90\\
30 & 35 & 40 & 28 & 40 & 36 & 42 & 50 & 20 & 22 & 25 & 18 & 65 & 73 & 58 & 79\\
30 & 35 & 40 & 28 & 40 & 36 & 42 & 50 & 20 & 22 & 25 & 18 & 65 & 73 & 58 & 79\\
58 & 63 & 38 & 60 & 39 & 40 & 39 & 41 & 148 & 130 & 162 & 138 & 88 & 93 & 98 & 108\\
58 & 63 & 38 & 60 & 39 & 40 & 39 & 41 & 148 & 130 & 162 & 138 & 88 & 93 & 98 & 108\\
50 & 55 & 60 & 48 & 60 & 56 & 62 & 70 & 40 & 42 & 45 & 38 & 85 & 93 & 78 & 99\\
50 & 55 & 60 & 48 & 60 & 56 & 62 & 70 & 40 & 42 & 45 & 38 & 85 & 93 & 78 & 99\\
80 & 85 & 60 & 82 & 61 & 62 & 61 & 63 & 170 & 152 & 184 & 160 & 110 & 115 & 120 & 130\\
60 & 65 & 70 & 58 & 70 & 66 & 72 & 80 & 50 & 52 & 55 & 48 & 95 & 103 & 88 & 109\\
85 & 90 & 65 & 87 & 66 & 67 & 66 & 68 & 175 & 157 & 189 & 165 & 115 & 120 & 125 & 135\\
80 & 80 & 90 & 78 & 90 & 86 & 92 & 100 & 70 & 72 & 75 & 68 & 115 & 123 & 108 & 129
\end{bmatrix}. \tag{23}$$

The handling t_{im}^{j} time is as follows:

$$t_{im}^{j}=\begin{bmatrix}
1.8 & 1.82 & 4.04 & 2.24 & 1 & 0.78 & 5.16 & 0.96 & 1.7 & 1.46 & 7.74 & +\infty & 1.1 & 1.26 & 6.45 & 0.9\\
0.9 & 1.90 & 3.67 & 1.84 & 0.5 & 0.81 & 4.68 & 0.76 & 0.85 & 1.52 & 7.02 & +\infty & 0.55 & 1.31 & 5.85 & 0.65\\
1.8 & 1.80 & 4.51 & 2.24 & 1 & 0.77 & 5.76 & 0.96 & 1.7 & 1.44 & 8.64 & +\infty & 1.1 & 1.25 & 7.2 & 0.9\\
0.9 & +\infty & +\infty & +\infty & 0.5 & +\infty & +\infty & +\infty & 0.85 & 0.85 & 0.85 & 0 & 0.55 & +\infty & +\infty & +\infty\\
1.44 & 1.33 & 1.97 & 2.08 & 0.8 & 0.57 & 2.52 & 0.88 & 1.36 & 1.07 & 3.78 & 2 & 0.88 & 0.92 & 3.15 & 0.8\\
1.8 & 1.33 & 2.16 & 2.24 & 1 & 0.57 & 2.76 & 0.96 & 1.7 & 1.07 & 4.14 & 2 & 1.1 & 0.92 & 3.45 & 0.9\\
1.08 & 1.35 & 2.07 & 1.92 & 0.6 & 0.58 & 2.64 & 0.8 & 1.02 & 1.08 & 3.96 & 2 & 0.66 & 0.94 & 3.3 & 0.7\\
0.9 & 1.37 & 2.16 & 1.84 & 0.5 & 0.58 & 2.76 & 0.76 & 0.85 & 1.10 & 4.14 & 2 & 0.55 & 0.95 & 3.45 & 0.65\\
2.16 & 2.48 & 5.08 & 2.4 & 1.2 & 1.06 & 6.48 & 1.04 & 2.04 & 1.98 & 9.72 & 2.5 & 1.32 & 1.72 & 8.1 & 1\\
2.16 & 2.48 & 5.17 & 2.4 & 1.2 & 1.06 & 6.6 & 1.04 & 2.04 & 1.98 & 9.9 & 2.5 & 1.32 & 1.72 & 8.25 & 1\\
2.16 & 2.46 & 5.36 & 2.4 & 1.2 & 1.05 & 6.84 & 1.04 & 2.04 & 1.97 & 10.26 & 2.5 & 1.32 & 1.70 & 8.55 & 1\\
2.16 & 2.46 & 5.08 & 2.4 & 1.2 & 1.05 & 6.48 & 1.04 & 2.04 & 1.97 & 9.72 & 2.5 & 1.32 & 1.70 & 8.1 & 1
\end{bmatrix}. \tag{24}$$

And, the loading conditions' satisfaction $u_3^{im,j}$ is as follows:

$$u_3^{im,j}=\begin{bmatrix}
0.5 & 0.75 & 1 & 0.5 & 0.5 & 0.75 & 1 & 0.5 & 0.5 & 0.75 & 1 & 0 & 0.5 & 0.75 & 1 & 0.5\\
1 & 1 & 0.5 & 1 & 1 & 1 & 0.5 & 1 & 1 & 1 & 0.5 & 0 & 1 & 1 & 0.5 & 1\\
0.5 & 0.75 & 1 & 0.5 & 0.5 & 0.75 & 1 & 0.5 & 0.75 & 0.75 & 1 & 0 & 0.5 & 0.75 & 1 & 0.5\\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\
0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0 & 0.75 & 1 & 1 & 0.75\\
0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0 & 0.75 & 1 & 1 & 0.75\\
0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0 & 0.75 & 1 & 1 & 0.75\\
0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0 & 0.75 & 1 & 1 & 0.75\\
1 & 0.75 & 0.5 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.5 & 0 & 1 & 0.75 & 0.5 & 1\\
1 & 0.75 & 0.5 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.5 & 0 & 1 & 0.75 & 0.5 & 1\\
1 & 0.75 & 0.5 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.5 & 0 & 1 & 0.75 & 0.5 & 1\\
1 & 0.75 & 0.5 & 1 & 1 & 0.75 & 0.75 & 1 & 1 & 0.75 & 0.5 & 0 & 1 & 0.75 & 0.5 & 1
\end{bmatrix}. \tag{25}$$

5.1. Example Results. Setting $\varepsilon = 10^{-8}$ and $\kappa = 10\%$, we calculate the satisfaction value matrices of $R_V' = (\lambda_{jim}')_{J\times I\times M}$ and $V_R = (\theta_{jim})_{J\times I\times M}$ are as follows:

$$R_v' = \begin{bmatrix} 0 & 0 & 0 & 0.0078 & 0.0087 & 0 & 0 & 0.005 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0149 \\ 0 & 0 & 0 & 0.0089 & 0.009 & 0 & 0 & 0.0053 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.017 \\ 0 & 0 & 0 & 0.0079 & 0.0083 & 0 & 0 & 0.0047 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0163 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0571 & 0 & 0 & 0 & 0 \\ 0.0511 & 0.0512 & 0.0366 & 0.0097 & 0.0108 & 0.0101 & 0.0027 & 0.0058 & 0.0594 & 0.0378 & 0.0164 & 0 & 0.0264 & 0.0172 & 0.0076 & 0.0222 \\ 0.0491 & 0.0502 & 0.0358 & 0.0095 & 0.0104 & 0.0099 & 0.0026 & 0.0056 & 0.0579 & 0.0373 & 0.0162 & 0 & 0.0256 & 0.0169 & 0.0074 & 0.0217 \\ 0.0476 & 0.0483 & 0.0347 & 0.0092 & 0.0095 & 0.0092 & 0.0025 & 0.0052 & 0.084 & 0.0458 & 0.0197 & 0 & 0.0237 & 0.0159 & 0.0071 & 0.0203 \\ 0.0456 & 0.047 & 0.0337 & 0.009 & 0.0091 & 0.009 & 0.0024 & 0.005 & 0.0791 & 0.0441 & 0.0191 & 0 & 0.0229 & 0.0156 & 0.007 & 0.0198 \\ 0.1125 & 0 & 0 & 0.0168 & 0.0239 & 0 & 0 & 0.0109 & 0.1302 & 0 & 0 & 0 & 0.0599 & 0 & 0 & 0.0463 \\ 0.137 & 0 & 0 & 0.019 & 0.027 & 0 & 0 & 0.012 & 0.191 & 0 & 0 & 0 & 0.0706 & 0 & 0 & 0.0526 \\ 0.1068 & 0 & 0 & 0.0162 & 0.0253 & 0 & 0 & 0.0114 & 0.1518 & 0 & 0 & 0 & 0.082 & 0 & 0 & 0.0589 \\ 0.1067 & 0 & 0 & 0.0162 & 0.027 & 0 & 0 & 0.012 & 0.1685 & 0 & 0 & 0 & 0.082 & 0 & 0 & 0.059 \end{bmatrix},$$

$$V_R = \begin{bmatrix} 0 & 0 & 0 & 0.8955 & 0 & 0 & 0 & 0.9367 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8924 & 0 & 0 & 0 & 0.933 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8498 & 0 & 0 & 0 & 0.7187 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8821 & 0 & 0 & 0 & 0 \\ 0 & 0.8446 & 0.8912 & 0.7103 & 0 & 0.9676 & 0.8266 & 0.8705 & 0 & 0 & 0.001 & 0 & 0 & 0.6314 & 0.5541 & 0 \\ 0 & 0.83 & 0.8885 & 0.6748 & 0 & 0.9439 & 0.8241 & 0.8341 & 0 & 0 & 0.001 & 0 & 0 & 0.5775 & 0.5525 & 0 \\ 0 & 0.7827 & 0.7444 & 0.6857 & 0 & 0.7223 & 0.6782 & 0.5563 & 0.3414 & 0.9424 & 0.0952 & 0.7811 & 0 & 0.2142 & 0.4707 & 0 \\ 0 & 0.7247 & 0.7402 & 0.6324 & 0 & 0.6248 & 0.6744 & 0.5034 & 0 & 0.9357 & 0.0952 & 0.6938 & 0 & 0.1085 & 0.4673 & 0 \\ 0.8928 & 0 & 0 & 0.9009 & 0.971 & 0 & 0 & 0.9896 & 0.3213 & 0 & 0 & 0.4957 & 0.7689 & 0 & 0 & 0.7171 \\ 0.897 & 0 & 0 & 0.9073 & 0.8303 & 0 & 0 & 0.7422 & 0.966 & 0 & 0 & 0.9727 & 0.5482 & 0 & 0 & 0.01 \\ 0.8795 & 0 & 0 & 0.9024 & 0.9827 & 0 & 0 & 0.9896 & 0.3868 & 0 & 0 & 0.5049 & 0.8092 & 0 & 0 & 0.7171 \\ 0.8629 & 0 & 0 & 0.9068 & 0.8302 & 0 & 0 & 0.7422 & 0.9489 & 0 & 0 & 0.9721 & 0.5519 & 0 & 0 & 0.01 \end{bmatrix}. \tag{26}$$

Assuming $\lambda' = 10^{-7}$ and $\theta' = 10^{-7}$ and using improved NIMP algorithm to solve the model, finally, we get the maximum weighted satisfaction of $F1$ is 0.7392, and the optimal matching result is shown in Table 4.

It can be seen from the result that the loading characteristics are considered in the allocation scheme so that the materials can be matched with particular vehicles. This scheme is more specific than the existing research and plays a more practical guiding role in material emergency transportation.

5.2. Example Analysis. Considering the uncertainty of arrival time in emergency transportation, we design the following experiments to show that the matching scheme still has certain reliability under different delay risks.

(1) Firstly, set $\kappa = 10\%$, and the delay probability ξ_j of V_j is given in Table 3. Assume in actual rescue, the disturbance coefficient κ_{actual} is 20%, and the delay time of the delayed vehicle is $\kappa_{\text{actual}} * T_{im}^j = 20\% * T_{im}^j$. In this condition, we calculate the satisfaction of the optimal scheme attacked by disturbance coefficient κ_{actual} (κ_{actual}-optimal scheme for short) and the scheme in Table 4 (κ-current scheme for short), respectively. Repeat this experiment 2000 times, and the comparison between the κ-current and the κ_{actual}-optimal scheme is shown in Figure 2. We use green circle which denotes κ-current scheme satisfaction and blue line which denotes κ_{actual}-optimal scheme satisfaction. It can be seen from Figure 2, when $\kappa = 10\%$ and disturbance coefficient κ_{actual} l is 20%, the gap between the κ-current scheme and the κ_{actual}-optimal scheme is not noticeable. This result shows that $\kappa = 10\%$ and can make the scheme stable to a certain extent.

(2) On the basis of the above experiment, we still make $\kappa = 10\%$ and change the value of disturbance coefficient κ_{actual} from 0 to 70%. Under each value of

Table 4: Information on vehicles.

Area	Vehicle type	Result
$R1$	Helicopter Freight train	$\{V_9\}\longleftrightarrow\{r_{11}\}$ $\{V_5\}\longleftrightarrow\{r_{12}, r_{13}, r_{14}\}$
$R2$	Helicopter Freight train	$\{V_{12}\}\longleftrightarrow\{r_{21}, r_{24}\}$ $\{V_8\}\longleftrightarrow\{r_{22}, r_{23}\}$
$R3$	Helicopter Freight train Ambulance	$\{V_{10}\}\longleftrightarrow\{r_{31}\}$ $\{V_7\}\longleftrightarrow\{r_{32}, r_{33}\}$ $\{V_4\}\longleftrightarrow\{r_{34}\}$
$R4$	Helicopter Freight train	$\{V_{11}\}\longleftrightarrow\{r_{41}, r_{44}\}$ $\{V_6\}\longleftrightarrow\{r_{42}, r_{43}\}$

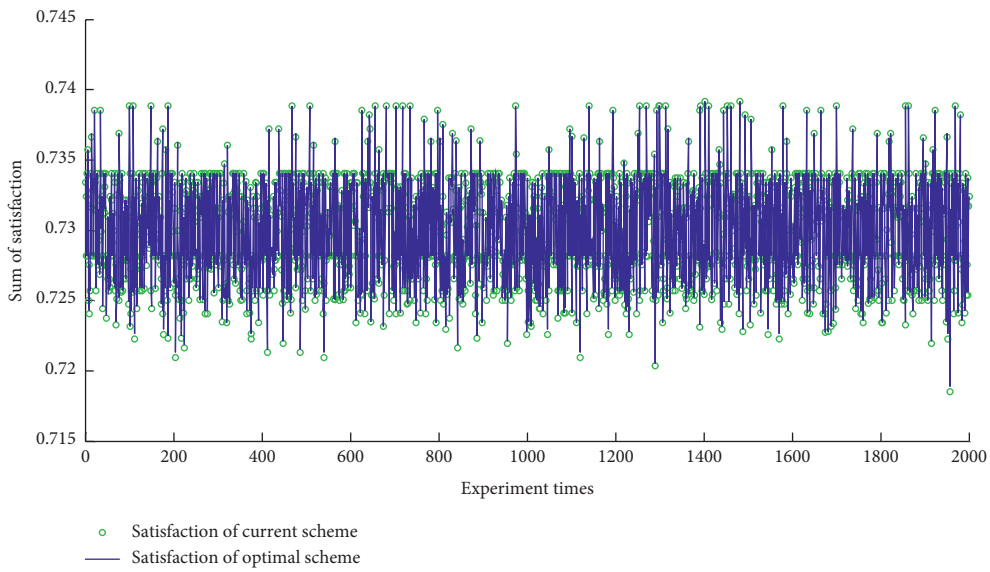


Figure 2: Comparison between the κ-current and the κ_{actual}-optimal schemes.

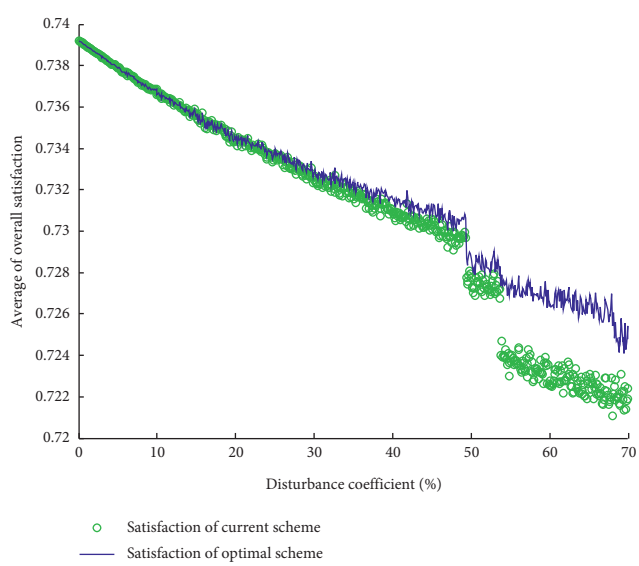


Figure 3: Relationship between the gap and the disturbance coefficient.

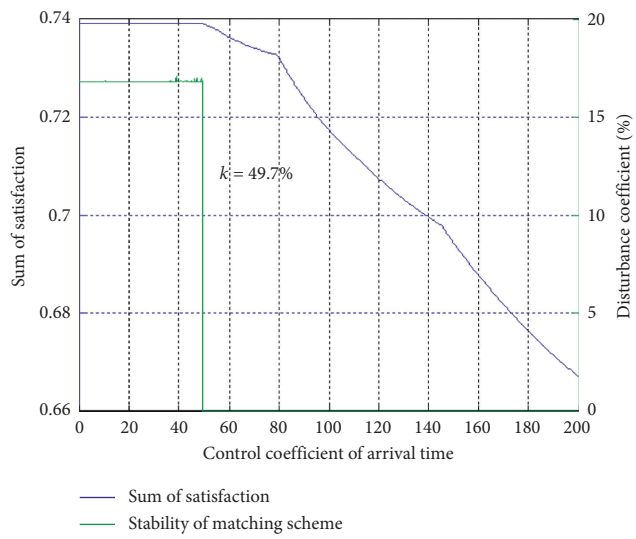


Figure 4: Relationship between κ and the satisfaction of schemes, and the relationship between κ and stability of the κ-current schemes.

κ_{actual}, we calculate the mean satisfaction value of the κ-current and κ_{actual}-optimal schemes. The comparison is shown in Figure 3.

We use the green circle which denotes the variation in κ-current scheme satisfaction and the blue line which denotes the variation in κ_{actual}-optimal scheme satisfaction. The result shows that, with the increase of the disturbance coefficient κ_{actual}, the average satisfaction of the κ-current schemes and the κ_{actual}-optimal schemes show a downward trend, and the descent speed of κ-current schemes is faster than κ_{actual}-optimal schemes. Even so, when the disturbance coefficient is less than 53.7%, the difference between the satisfaction of the κ-current schemes and the κ_{actual}-optimal schemes is still not significant. When the disturbance coefficient exceeds 53.7%, because the overall satisfaction of $R3$ decrease significantly, the gap between the κ-current schemes and the κ_{actual}-optimal schemes has been widened.

(3) The value of κ not only affects the reliability of the scheme but also affects the satisfaction of the κ-current scheme, so we change κ from 0 to 200% to find a reasonable value. Figure 4 shows the relationship between κ and the satisfaction of the κ-current scheme (green line) and the relationship between κ and stability of matching scheme (blue line) as well.

The green line expresses if $\kappa \leq 49.7\%$, and the satisfaction of the κ-current schemes remains unchanged. Once $\kappa > 49.7\%$, the satisfaction of κ-current schemes falls rapidly.

The blue line shows the value of disturbance coefficient κ_{actual} that the satisfaction of the κ-current scheme is lower than that of the optimal scheme for the first time. When $\kappa \leq 49.7\%$, the value remains almost unchanged. When $\kappa > 49.7\%$, because of the nonideal κ-current scheme, all the optimal schemes are better than the κ-current ones.

Based on the above analysis, the value of κ should not exceed 49.7%.

In practical application, the value of κ can also be analyzed according to the specific situation of vehicles and materials to let the scheme have better stability and operability.

6. Summary

This paper studies the matching problem between the materials to be transported and the comprehensive transportation network emergency vehicles in the process of rescue, aiming to promote the information sharing of transportation resources and the rational allocation of transportation resources and improve the quality of emergency rescue. We first establish evaluation index systems to measure the satisfaction of both parties. Next, based on the realistic constraints of the materials and vehicles matching, the multiobjective optimization model is established to maximize the satisfaction of the matching parties. Then, an improved NIMP algorithm is designed to solve the model. Finally, through the calculation of an example, it shows that the proposed method can obtain the optimal matching scheme different from the existing literature, which can specify the materials transported by certain vehicles on the basis of loading and arrival time demand. At the same time, repeated experiments show that the obtained matching scheme has excellent stability in the face of the uncertainty of transportation time caused by emergencies.

The limitations of the paper are as follows. (1) The situation of multimodal transportation is not considered. (2) It is not applicable that materials can be separated. (3) Discussion about blending loading demand for materials is

lacking. Therefore, in the next step of research, it is attemptable to transform the one-to-many matching mode into many-to-many matching mode and improve the satisfaction index extraction based on the characteristics of emergency.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Key R&D Program of China (2017YFB1200702), National Natural Science Foundation of China (Project no. 52072314), Sichuan Science and Technology Program (Project nos. 2020YFH0035, 2020YJ0268, 2020YJ0256, and 2020JDRC0032), Chengdu Science and Technology Plan Research Program (Project nos. 2019YF0501493SN and 2020RK0000036ZF), Natural Science Foundation of Zhejiang Province, China (LQ18G030012), and Humanities and Social Sciences Fund of Ministry of Education, China (18YJC630190).

References

[1] L. Y. Zhang, T. Fei, J. Zhang, and J. Li, "Application of emergency logistics distribution routing optimization based on improved ant colony algorithm," *Advanced Materials Research*, vol. 268–270, pp. 1726–1732, 2011.

[2] Z.-H. Hu, "A container multimodal transportation scheduling approach based on immune affinity model for emergency relief," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2632–2639, 2011.

[3] L. Özdamar and O. Demir, "A hierarchical clustering and routing procedure for large scale disaster relief logistics planning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 3, pp. 591–602, 2012.

[4] C. G. Rawls and M. A. Turnquist, "Pre-positioning and dynamic delivery planning for short-term response following a natural disaster," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 46–54, 2012.

[5] A. Bozorgi-Amiri, M. S. Jabalameli, and S. M. J. Mirzapour Al-e-Hashem, *A Multi-Objective Robust Stochastic Programming Model for Disaster Relief Logistics Under Uncertainty*, Spring-Verlag, New York, NY, USA, 2013.

[6] A. Afshar and A. Haghani, "Modeling integrated supply chain logistics in real-time large-scale disaster relief operations," *Socio-Economic Planning Sciences*, vol. 46, no. 4, pp. 327–338, 2012.

[7] F. Sabouhi, M. Heydari, and A. Bozorgi-Amiri, "Multi-objective routing and scheduling for relief distribution with spilt delivery in post-disaster response," *Journal of Industrial and Systems Engineering*, vol. 9, no. 3, pp. 17–27, 2016.

[8] L. Huang, J. Yang, and C.-H. Chen, "An improved swarm intelligence algorithm for multi-Item joint ordering strategy of cruise ship supply," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5048629, 9 pages, 2020.

[9] G. Laporte, F. Louveaux, and H. Mercure, "Models and exact solutions for a class of stochastic location-routing problems," *European Journal of Operational Research*, vol. 39, no. 1, pp. 71–78, 1989.

[10] G. Barbarosoglu and Y. Arda, "A two-stage stochastic programming framework for transportation planning in disaster response phase," *Journal of the Operational Research Society*, vol. 55, no. 1, pp. 43–53, 2004.

[11] X. Duan, S. Song, and J. Zhao, "Emergency vehicle dispatching and redistribution in highway network based on bilevel programming," *Mathematical Problems in Engineering*, vol. 2015, Article ID 731492, 12 pages, 2015.

[12] W. Klibi, S. Ichoua, and A. Martel, "Prepositioning emergency supplies to support disaster relief: a case study using stochastic programming," *Information Systems and Operational Research*, vol. 1, no. 56, pp. 50–80, 2018.

[13] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[14] A. E. Roth, "Common and conflicting interests in two-sided matching markets," *European Economic Review*, vol. 27, no. 1, pp. 75–96, 1985.

[15] A. E. Roth, "On the allocation of residents to rural hospitals: a general property of two-sided matching markets," *Econometric a*, vol. 54, no. 54, pp. 425–427, 1986.

[16] D. He, C. Cui, and J. Guo, "Research on the selection of military emergency material reserve," *Mathematics in Practice and Theory*, vol. 49, no. 15, pp. 61–68, 2019.

[17] J. Li, X. Yao, D. Chen, and S. Ni, "Evaluation index selection of comprehensive inter-regional traffic emergency rescue plan," *China Safety Science Journal*, vol. 28, no. S2, pp. 185–190, 2018.

[18] D. Chen, Y. Sun, J. Li, and S. Ni, "Construction of evaluation index system for emergency rescue capacity of rail transit under serious epidemic situation," *Journal of Traffic and Transportation Engineering*, vol. 20, no. 3, pp. 129–138, 2020.

[19] L. Su, C. Yin, D. Chen, H. Lv, and Q. Zhang, "Cascading failure in multiple critical infrastructure interdependent networks of syncretic railway system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021.

[20] T. Yang, X. Peng, D. Chen, F. Yang, and M. Muneeb Abid, "Research on trans-region integrated traffic emergency dispatching technology based on multi-agent," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 5763–5774, 2020.

[21] J. Zhu, R. Wang, and Y. Li, "A decision method of vehicle cargo bilateral matching based on the information of uncertainty language relevance," *Journal of Systems Science*, vol. 26, no. 1, pp. 86–91, 2018.

[22] Z. Wang, Y. Li, F. Gu, J. Guo, and X. Wu, "Two-sided matching and strategic selection on freight resource sharing platforms," *Physica, A. Statistical Mechanics and its Applications*, vol. 559, pp. 1–18, 2020.

[23] B. Yang, X. Ye, R. Wang, and B. Shuai, "Method for vehicle-cargo two-sided fair matching based on intuitionistic Fuzzy optimization," *Computer Integrated Manufacturing Systems*, pp. 1–14, 2021.

[24] B. Wang, J. Cui, and D. Kong, "Research on the decision-making method of vehicle-cargo two-sided matching based on grey absolute incidence degree," *Value Engineering*, vol. 38, no. 23, pp. 122–125, 2019.

[25] Y. Gu, M. Su, L. Zhang, and Y. Zheng, "Research on vehicle and goods attributes matching of platform based on bidding

and order distribution mode," *Journal of Wuhan University of Technology (Transportation Science &. Engineering)*, vol. 44, no. 3, pp. 450–455, 2020.

[26] Z.-S. Chen, X.-L. Liu, K.-S. Chin, W. Pedrycz, K.-L. Tsui, and M. J. Skibniewski, "Online-review analysis based large-scale group decision-making for determining passenger demands and evaluating passenger satisfaction: case study of high-speed rail system in China," *Information Fusion*, vol. 69, pp. 22–39, 2021.

[27] Z.-S. Chen, X.-L. Liu, R. M. Rodriguez et al., "Identifying and prioritizing factors affecting in-cabin passenger comfort on high-speed rail in China: a fuzzy-based linguistic approach," *Applied Soft Computing*, vol. 59, pp. 1–18, 2020.

[28] A. E. Roth, "The evolution of the labor market for medical interns and residents: a case study in game theory," *Journal of Political Economy*, vol. 92, no. 6, pp. 991–1016, 1984.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5581984, 12 pages
https://doi.org/10.1155/2021/5581984

Research Article

2.5D Facial Personality Prediction Based on Deep Learning

Jia Xu,[1,2] **Weijian Tian**,[1] **Guoyun Lv**,[1] **Shiya Liu**,[3] **and Yangyu Fan**[1]

[1]*School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China*
[2]*North China University of Science and Technology, Tangshan, China*
[3]*Content Production Center of Virtual Reality, Beijing, China*

Correspondence should be addressed to Guoyun Lv; lvguoyun101@nwpu.edu.cn

Received 1 February 2021; Revised 14 May 2021; Accepted 16 June 2021; Published 30 June 2021

Academic Editor: Chunjia Han

The assessment of personality traits is now a key part of many important social activities, such as job hunting, accident prevention in transportation, disease treatment, policing, and interpersonal interactions. In a previous study, we predicted personality based on positive images of college students. Although this method achieved a high accuracy, the reliance on positive images alone results in the loss of much personality-related information. Our new findings show that using real-life 2.5D static facial contour images, it is possible to make statistically significant predictions about a wider range of personality traits for both men and women. We address the objective of comprehensive understanding of a person's personality traits by developing a multiperspective 2.5D hybrid personality-computing model to evaluate the potential correlation between static facial contour images and personality characteristics. Our experimental results show that the deep neural network trained by large labeled datasets can reliably predict people's multidimensional personality characteristics through 2.5D static facial contour images, and the prediction accuracy is better than the previous method using 2D images.

1. Introduction

There has been a long history of attempts to assess personality based on facial morphological features [1], a practice known as physiognomy. Of course, it is not only the East, but people of all ages all over the world have studied this field, and their attitudes are mixed. Aristotle once proposed that facial features can reflect personality characteristics to some extent. Personality is a kind of psychological structure in which a few stable and measurable individual characteristics are used to explain people's different behaviors [2]. Today, there is increasing research interest on the relationship between facial images and personality prediction. Studies by Todorov et al. [3–5] and others have shown that experts can reliably infer a person's personality traits from his facial appearance and use it to track crimes, campaigns, and medical care.

To date, there have been enough researches on character recognition by machine learning worldwide. In [6], the authors studied and proposed the key facial features that have an import impact on people's first impression. We can draw at least four valid inferences from other people's facial features [7]. Reference [8] examined the relationship between self-reported personality traits and first impressions. To investigate whether a computer can learn to assess human traits, the authors of [9] used a machine learning method to construct automatic feature predictors based on facial structure and appearance descriptors and found that all personality traits analyzed were accurate predictors. Recent studies on the static features of human face suggested that certain areas have evolved to play an important role in social communication [10] and that individuals with higher facial attractiveness personality traits have higher mating success [11].

Previous studies, either based on the five-factor model of personality or based on the Big Five (BF) model, have found there is a certain relationship between facial images and general personality characteristics. However, a cursory assessment of the predictions and results of these studies will reveal much controversy, with research results seemingly inconsistent and difficult to replicate [10] (see Table 1 for existing research results). These inconsistencies may be due

to the use of small numbers of stimulation schemes or to very large differences in the corresponding methods. Existing research datasets are insufficient, and most current research on face feature extraction has prioritized 2D faces, especially from front views [23]. However, a reliance only on 2D positive images causes much valuable information related to personality to be lost [24]. Prominent facial areas, such as prominent landmarks of the forehead, nose, and chin, are related to a person's personality [25]. In fact, positive and lateral facial expressions are naturally complementary. Therefore, multiple perspectives (front, side, and 2.5D) facial images are more likely to describe a person's personality comprehensively and accurately. Herein, we use the term "2.5D" to refer to combinations of front and side views.

The two main topics in existing face personality prediction research are the acquisition of datasets (face photos and personality data) and the design of computing networks.

1.1. Construction of Face Database. The establishment of the face database plays a vital role in verifying our model and ensuring its ability of generalization. Ideally, the face database should contain face samples from people of different genders, races, and ages, displaying different personality traits. However, to date, there is no such database for automatic personality calculation. In fact, differences in research backgrounds often result in the creation of independent databases based on their specifics situations; consequently, the number, age, gender composition, expression, race, and posture of existing facial image samples are not identical. A summary of the facial image databases constructed in the existing face-based personality computing research can be seen in Table 2.

1.2. Selection of the Personality Evaluation Model. Intuitively, evaluating a person's personality traits involves learning how to choose the adjectives from trait theory to describe it accurately. In all the literature on automatic personality prediction, two ways to evaluate a person's personality traits are described: (1) self-evaluation and (2) evaluation by others. Completion of the personality assessment scale in the first person, that is, a self-assessment, is traditionally considered to produce a person's real personality [22]. Completion of the questionnaire in the third person (e.g., substituting "this person tends to be sociable" for "I'm sociable") leads to attribution and results that are evaluated by others. Under the evaluation of others, each topic must be evaluated by several evaluators, and each evaluator must evaluate all the participants in the experiment. Statistical criteria such as the reliability in [27] allow the number of assessors to be set according to the agreement of both parties.

The theory of personality traits states that personality traits are an effective characteristic of individual behavior, an effective component of an individual, and a basic unit commonly used to evaluate personality. Common theories about traits include Allport's trait theory (common traits and personal traits), Cattell's theory of personality trait (in which everyone has 16 traits), Eysenck's three-factor model (extroversion, psychoticism, and neuroticism), Tapperth's five-factor model (commonly known as the Big Five: extroversion, agreeableness, sense of responsibility, neuroticism, and openness), and Terrigen's seven-factor model (positive emotionality, negative potency, positive potency, negative emotionality, reliability, agreeableness, and heredity). In face-based personality computing, one important element is the selection of the appropriate trait theory model.

In existing research on personality prediction based on facial features, the methods used to evaluate personality are as shown in Table 3 below.

1.3. Selection of the Prediction Network. In recent studies on facial personality, many methods have been adopted, such as the Parzen window [29], decision tree [30], naive Bayes [31], kNN, [31] and random forest [32]. Rojas et al. [14, 15] conducted a classification experiment using the most advanced classifier. Zeng [17] used a deep confidence network classification algorithm based on the backpropagation (BP) algorithm. Brahnam and Nanni [20] used principal component analysis (PCA) and random combinations of training and testing sets to train and test their models with 20 repetitions for each personality feature dimension. Kachur et al. [16] proposed a computer vision neural network (NNCV). Methods used for personality prediction in different studies are shown in Table 4.

The aim of this study is to investigate the association between facial image cues and self-reported Big Five personality traits by training a series of neural networks to predict personality traits in static face images. In view of the problems in previous studies, the contributions and innovations of this paper include the following: firstly, a large dataset composed of facial photos and personality characteristics is constructed. The dataset contains 13,347 pairs of data, 360 of which were collected from facial profile images, and a 2.5D dataset was constructed to obtain a more comprehensive understanding. Secondly, an improved deep learning algorithm was used to predict personality characteristics, which reduced the requirements from previous research on the quality of the face images; it was expected that a complex deep learning algorithm could be used to capture face images under uncontrolled conditions. Thirdly, the changing trend of facial characteristics of Asian college students with five personality dimensions ranked from high to low was predicted.

The experimental results of this paper show that, on the one hand, we can reliably predict some personality traits using static facial images; on the other hand, the performance of the facial feature extraction model in predicting personality based on 2.5D images is better than that of 2D images.

The rest of this paper is organized as follows. In Section 2, we describe the creation of our own dataset, including the face dataset and the Big Five personality assessment result dataset. In Section 3, we predict the personalities of positive faces based on an improved deep learning network and the changes in the average face from low to high. The method

Table 1: Summary of the literatures on the accuracy of personality prediction.

References	*N* photo	Assessment	Trait	Highest accuracy (%)
[12, 13]	186	Self-report	Conscientiousness (M)	81.56
			Conscientiousness (F)	82.22
			Skepticism (M)	72.64
			Skepticism (F)	82.22
[14, 15]	66	Other report	Dominance	91.23
			Threatening	90
			Extraversion	
[7]	244	Self-report	Openness, striving, and domination (F)	63
			Reliability, friendliness, and responsibility (M)	65
[16]	12,447	Self-report	Big Five (BF)	58
[17]	608	Self-report	Neuroticism	82.35
			Extraversion	84.31
			Rigorism	84.31
[18]	829	Other report	Neuroticism, openness, and extraversion	65
[19]	1856	Other report	Criminality	89.51
[15]	220	Other report	Warmth	76
			Reliability	81
[20]	480	Other report	Intelligence, maturity, sociality, dominance, warmth, and credibility	80
[19]	1856	Other report	Criminality	89.5
[21]	10,000	Other report	BF	89.11
[22]	10,000	Other report	BF	90.94

Note. The table reflects the prediction accuracies of existing studies on personality prediction with images of the whole face. *N* photo is the number of samples, assessment is the personality assessment method of the study, M is the result for images of males, and F is the result for images of females.

Table 2: Attributes of face datasets in existing studies.

References	Number of samples	Age	Gender	Race	Posture	Expression	Data source
[12, 19]	186	18–22	M-F	Asian	Front	Neutral	Students of Xiamen Institute of Technology (Arts and Science)
[13, 14]	66	20–30	M-F	White	Front	Neutral	Amateur actor face database from Karolinska [26]
[25]	244	18–37	M-F	White	Front	Neutral	Danish University of Technology Campus recruitment tester
[7]	608	18–22	M-F	Asian	Front	Neutral	Undergraduates of different disciplines and grades in a university in Jiangxi Province
[17]	829	20–39	M-F	Varied	Front	Neutral	Color-FERET
[24]	66	20–30	M-F	White	Front	Neutral	Amateur actor face database from Karolinska [26]
[13]	650	30–50	M-F	Varied	Front	Neutral	Images of real politicians
[18]	1856	18–55	M-F	Asian	Front	Neutral	Two subsets separately containing images of criminals and noncriminals
[15]	3998	18–25	M-F	Varied	Front	Neutral	Images downloaded from a social network
[19]	220	Varied	M-F	Varied	Front	Neutral	"FACES" software synthesis
[16]	12,447	Varied	M-F	White	Front	Neutral	Volunteers' self-photos
[20]	480	Varied	M-F	Varied	Front	Neutral	"FACES" software synthesis
[21]	5563	Varied	M-F	Varied	Blend	Neutral	Video collection: the ECCV ChaLearn LAP 2016 competition
[22]	10,000	Varied	M-F	Varied	Blend	Neutral	Video collection: the ECCV ChaLearn LAP 2016 competition

and experimental results of personality prediction with our model based on 2.5D face images are presented and discussed in Section 4. Finally, in Section 5, we analyze and give some examples of the applicability of the research results.

2. Dataset and Preprocessing

2.1. Samples and Procedure. The official language used in this study is Chinese. Participants were anonymous college student volunteers recruited by the research group through advertisements on the social network pages of colleges and universities. The data were based on a sample of 5,560 male and 8,547 female college students aged 18 to 25 (some face photos are shown in Figure 1). They were not paid financially but were given a free report on their Big Five personality traits. The data required for the experiment (face pictures and personality scores) were collected online through a dedicated personality research website and a mobile application. The participants signed and submitted an informed consent form, completed a five-person personality

Table 3: Personality evaluation models in personality prediction based on facial features.

References	Personality assessment	Personality trait theory	Feature expression
[12, 19]	Self-report	16 personality factors (16PF)	Score [0, 9]
[13, 14]	Other report	Artificially selected personality descriptors	14-dimensional personality score
[25]	Self-report	Big Five (BF)	Score [0, 9]
[7]	Self-report	BF	Score [1, 60]
[17]	Other report	BF	Score [0, 9]
[24]	Other report	Artificially selected personality descriptors	Score [0, 9]
[13]	Other report	Artificial selection: dominance, attraction, credibility and extraversion	Score
[18]	Other report	Artificial selection: dominance, warmth, sociality and credibility	Score [0, 9]
[15, 20]	Other report	Artificial selection: intelligence, maturity, warmth, sociality, dominance and credibility	Score [1, 3]
[16]	Self-report	BF	Score [0, 60]
[18]	Self-report	Eysenck's personality questionnaire—revised (EPQ-R)	Score [0, 120]
[28]	Other report	Eysenck's three-factor model	Score
[21]	Other report	BF	Score [0, 1]
[22]	Other report	BF	Score [0, 1]

Table 4: Personality prediction algorithms used in different studies.

Literature	Publication year	Algorithm
[12, 13]	2011	Parzen window, decision tree, naive Bayesian, kNN, and random forest
[10, 11]	2018	Deep learning network
[23]	2014	Deep confidence network based on the backpropagation algorithm
[24]	2018	Support vector machine (SVM)
[13]	2010	SVM (RankSVM)
[14]	2010	Logistic regression, kNN, SVM, and CNN
[20]	2016	SVM
[14, 33]	2018	SVM and CNN
[16]	2020	Computer vision neural network (NNCV)
[29]	2018	Parzen window
[30]	2018	Decision tree
[31]	2001	k-nearest neighbour and naïve Bayes
[32]	2001	Random forest
[21]	2016	Deep bimodal regression (DBR)
[22]	2016	DCC, UCAS, and BU-NKU

questionnaire, filled in their age, gender, and major, and uploaded frontal photos that showed a neutral, unsmiling expression and to avoid thick facial makeup and other decorations, such as hats. To study the contribution of a person's profile to personality prediction, we also collected pictures of the profiles of an additional 360 students.

2.2. Ethical Approval. Participants were required to agree in writing to participate in the study, and their data was collected only after obtaining their authorization. In addition, we anonymously collected self-reported personality assessment data by assigning a number to each participant. Furthermore, the face and personality data were only used for scientific research, and no personal data will be disclosed to the outside world.

2.3. Establishment of the Personality Dataset (Big Five Personality Traits). To study the contribution of the profile view to personality prediction, we collected profile views from an additional 360 students. We expended much effort to collect personality trait data from the participants. Research and experimental results over the years have shown that the same behavioral characteristics appear in various environments and cultures with surprising regularity, indicating that they actually correspond to certain similar personality psychological phenomena [18]. Today, the Big Five is considered one of the most dominant and influential models of personality research [28]. This article uses the BF model. The Big Five are openness, conscientiousness, extraversion, agreeableness, and neuroticism. Each dimension is like a ruler, and the personality characteristics of each tester will fall in a certain position of each ruler. The closer this point lies to the end point of the ruler, the greater preference the user has toward the corresponding personality trait. A score of 0 to 60 is set for each dimension, such as agreeableness, where the higher the individual's score, the more easygoing and pleasant the personality is [21, 22].

Questionnaires based on a Likert scale are the most commonly used tools for scoring the BF dimensions [27]. The most popular items include the revised NEO-PI-R (240 items) [34], the NEO-FFI (60 items) [35], and BFI (44 items)

Figure 1: Front view images of some Asian college students.

[36] (see [2] for an extensive investigation). By retaining only the items that are most relevant to the results of the whole document [26, 37], a shorter questionnaire (60 items) can be established that can be filled much faster (the 60 questions are given in Supplementary Materials).

First-person questionnaires such as those in Annex 1 lead to self-assessment, which is traditionally considered to produce a person's real personality [14]. For self-assessment, the biggest limitation is that the subject may tend to bias her score toward the characteristics of social expectations, especially when the assessment may have negative results, such as failing an interview. As a result, statements such as "I tend to be lazy" may be rated as disagreeable because the respondent will attempt to convey positive impressions and hide negative features. However, a large number of experiments have shown that the self-assessment results are highly correlated with the evaluations of others provided by familiar observers (spouses, family members, etc.) [33]. This proved to be an important step in accepting the questionnaire as a method of personality evaluation. Therefore, we also used numerous self-assessments in the experiment.

5560 men and 8547 women completed the personality assessment questionnaire and uploaded 14021 photos. After final verification was performed and the face and personality data were merged, the dataset included 13,347 valid questionnaires and 13,347 related photos (see below). Participants ranged in age from 18 to 25, with an average age of 21.4 years for females, accounting for 62.1% of the total, and 20.7 years for males, accounting for 37.8%. We randomly divided the dataset into training dataset, test dataset, and verification dataset, accounting for 90%, 5%, and 5% of the total dataset, respectively. In addition, we randomly collected side face photos of another 360 participants to study the contribution of 2.5D faces to personality prediction.

2.4. Screening and Analysis of Image and Personality Data. Each participant was given scores on five personality trait dimensions based on the Big Five personality test, each of which was scored as a discrete number between 1 and 60. We use the tripartite method to divide the personality scores of different dimensions into "low, medium, and high," such as low neuroticism, medium neuroticism, and high neuroticism. The statistical results are shown in Table 5. For the sake of simplicity, the letters O, C, E, A, and N are used to denote openness, conscientiousness, extraversion, agreeableness, and neuroticism, respectively.

This result is basically in line with the personality characteristics of Asians, who are considered to be relatively conservative, kind-hearted, and introverted. Therefore, in our dataset, people with high openness and high neuroticism accounted for a relatively small proportion. To facilitate calculation and analysis, we classified the personality characteristics of the population into two additional categories: "not obvious" and "obvious" according to the data collected after the survey. Although the data were classified into "high, medium, and low" at first, because the proportions of participants with high neuroticism, high extraversion, and high openness were very low almost to the point of negligibility, we divided these small numbers of participants into the next closest categories (the final classification is shown in Table 6).

We applied the functions of face and eye detection, alignment, resizing, and clipping provided by Dlib library (dlib.net website) to process the face images and obtained a group of normalized images with the pixel size of 112×112.

By matching the answers to the questionnaire with the face photos one by one, we were able to obtain a valid set of Big Five questionnaires and images, totaling 13,347 pairs.

3. Neural Network for Personality Prediction Based on 2D Images

Previous studies on personality prediction were conducted by means of artificial feature collection, which could result in the loss of personality-related features [12, 14, 20, 38–41]. We predict that personality characteristics will be reflected in the person's entire facial image (including the profile) rather than in a certain number of isolated facial features. Consequently, we employed a deep learning method to extract high-level features from face images for personality prediction. We used MobileNetV2 and residual network version 50 (ResNet50), two deep learning networks that are popular in academia, to classify personality traits. Then, an improved personality prediction network—Soft Threshold-Based Neural Network for Personality Prediction (S-NNPP)—was proposed. To verify the experimental results, 5-fold cross-validation method was used. The data were randomly scrambled and divided into five pieces, and for each fold, one piece of data was further divided into equally sized test and validation sets, and the remaining four pieces as the training set. Take the average of the verification results from the five folds as the final result. In the training process, focus loss, data enhancement, upsampling, and cost sensitivity were introduced to solve the problem of sample imbalance. All training in this section was fine-tuned based on the ImageNet pretrained model with stochastic gradient descent as our optimization strategy. At the beginning of training, we set the learning rate to 0.001 and adopted the

Table 5: Three-category personality data classification.

Category	Trait (number of people)				
	O	C	E	A	N
Low	6328	5043	18	7934	29
Medium	7019	8226	10,081	5413	12,861
High	0	78	3248	0	457
Total	13,347	13,347	13,347	13,347	13,347

Table 6: Two-category personality data classification.

Category	Trait (numbers)				
	Neuroticism	Extraversion	Openness	Agreeableness	Conscientiousness
Not obvious	12,861	10,081	6328	7934	5043
Obvious	486	3266	7019	5413	8304
Total	13,347	13,347	13,347	13,347	13,347

ReduceLROnPlateau, which can adjust the learning rate dynamically according to the loss, as the learning rate optimization strategy.

3.1. Soft Threshold-Based Neural Network for Personality Prediction (S-NNPP). Recent studies have shown that networks based on attention mechanisms can achieve good performance in classification tasks. Therefore, an increasing number of networks add various attention operations in ResNet. The ResNeSt proposed in 2020 can be regarded as an "integrated master" that incorporates the best of the previous versions of ResNet. Based on the in-depth analysis of GoogLeNet, the selective kernel network (SKNet), and the squeeze-and-excitation network (SENet), a deep neural network called S-NNPP was designed. The objective is to select a network architecture for multiscale image feature extraction and to achieve good image classification performance. We introduced the multipath mechanism of GoogLeNet and the feature map attention module of SKNet in ResNeSt. We also introduced channel attention by adaptively recalibrating the channel characteristic response, following the architecture of SENet. Due to the outstanding performance of ResNeSt in image classification, we employed its basic modules to make subsequent network improvements.

The network diagram in Figure 2 shows that in the ResNeSt block, the $3 * 3$ convolution in ResNet was replaced by grouping convolution through splitting, and attention was paid through multiple branches. Here, grouped convolution was used in every path. Finally, following the softmax operation, the convolution results of each group were merged.

Personality data are easily labeled as fuzzy, so in this study, we introduce soft thresholding [42] to improve the model's adaptability to noisy data. In many signal denoising methods, soft thresholding was the core step. It is used to set the feature whose absolute value is below a certain threshold value to 0 and adjusting other features accordingly—that is, to perform shrinkage. Here, the threshold is a parameter that must be set in advance, and its value has a direct impact on the noise reduction results. In terms of soft thresholding operation, the input-output relationship is shown in Figure 3.

The soft-thresholding formula is as follows:

$$\operatorname{soft}(x, T) = \begin{cases} x + T\,(x \leq -T), \\ 0\,(|x| \leq T), \\ x - T\,(x \geq T), \end{cases} \tag{1}$$

where $|x|$ is the wavelet transform coefficient and T is the preselected threshold.

It can be seen from the formula and the figure that the soft threshold function removes features whose absolute value is less than the threshold T and shrinks the features whose absolute value is greater than the threshold toward 0. When applied to the network, it can compress and retain the important features and filter the unimportant features. Due to the influence of various factors, the redundant information contained in different samples tends to be different, so different thresholds need to be set for different samples. Therefore, when performing soft threshold segmentation on the feature maps, we added a subnetwork to the basic network to automatically learn a set of thresholds. In this way, a unique set of soft thresholds can be obtained for each sample to remove redundant information. The adjusted basic network module is shown in Figure 4.

Figure 5 shows the network architecture of the soft threshold block.

In general, the Soft_ResNeSt network consists of two paths, one taking the entire image as input, and the other taking only the face region, which was obtained by an open-source OpenCV face region extractor. The improved ResNeSt module described above was used in the basic module of the two paths, and then according to a weighted parameter α, the prediction results were fused. Figure 6 shows the overall structure of our network.

In this section, the traditional BP network, two kinds of deep learning networks, and the improved S-NNPP network are compared in terms of their performance in personality prediction. Among them, the classification results of the lightweight MobileNetV2 for the personality data are good for neuroticism and extroversion, while the effect of

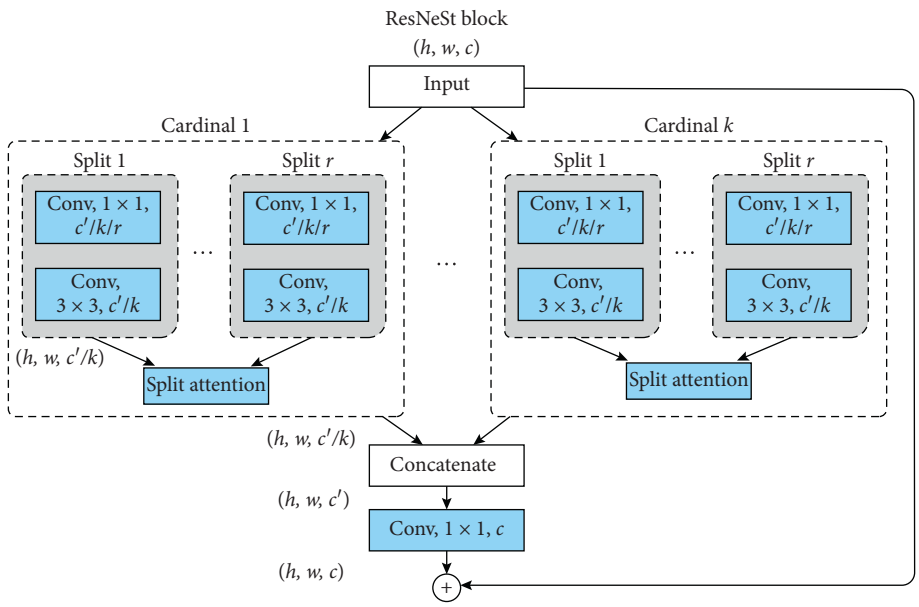


Figure 2: Network structure of ResNeSt.

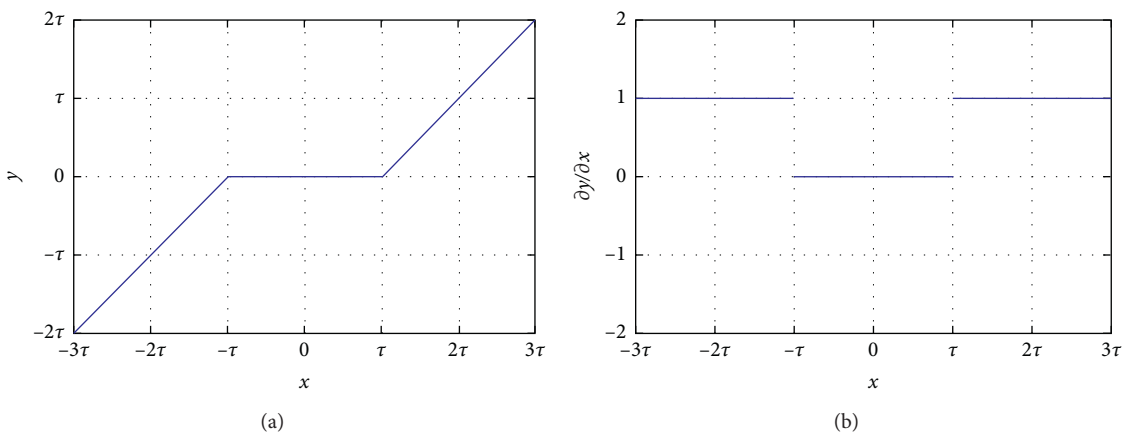


Figure 3: Soft-thresholding curves. The soft threshold function sets the input data with an absolute value lower than this threshold to zero, and the input data with absolute value greater than this threshold is also shrunk towards zero, and the relationship between input and output is shown in (a); the partial derivative of output y of the soft threshold function with respect to input x in (b).

openness, pleasantness, and responsibility is not obvious. Comparatively, the results of the complex network ResNeSt50 were slightly improved, indicating that the complicated network architecture can better extract depth characteristics related to personality. Finally, combining ResNeSt and soft-threshold technology, we generated S-NNPP, an improved personality prediction network, which is substantially better than MobileNetV2 and ResNeSt50 in predicting five personality dimensions.

3.2. Results and Discussion. In this study, the data were scrambled and randomly divided into five sections, one of the sections was further divided into equally sized test and validation sets, and the remaining four sections served as the training set. The verification data we used were from an independent verification dataset, which contained the predicted scores of 1335 facial images of 1335 volunteers. The final prediction result is the average of the verification results from using each of the five parts as the verification set.

We tested the accuracy of different neural networks in predicting five personality traits. The true and false positive rates and F1 scores of the three deep learning networks in predicting the five traits are shown in Table 7. Neuroticism and extroversion were significantly easier to identify than others, as indicated by a recognition rate of over 90% (see Figure 7: receiver operating characteristic (ROC) curves). The degree of recognition openness, agreeableness, and conscientiousness for the three networks is relatively weak but better than that by the line representing chance; this is different from the existing conclusions to some extent [43, 44]. There are several reasons why our research results may be different from other results. Firstly, all our volunteers were Asian, who, due to cultural differences, emphasize their

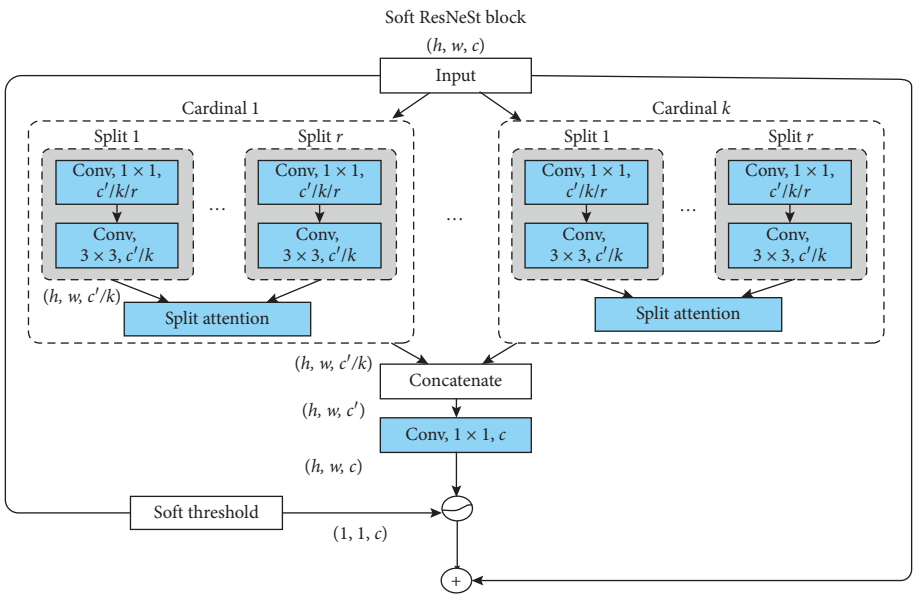
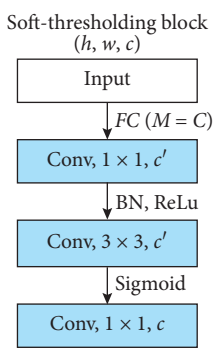


Figure 4: Basic network module with soft threshold.



The small network is used to obtain a set of thresholds.

Figure 5: Network architecture of the soft-thresholding block.

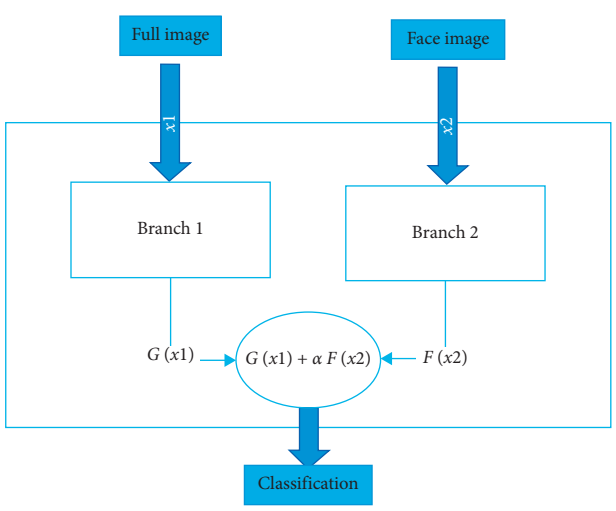


Figure 6: Overall network structure of S-NNPP.

"openness," "easygoing nature," and "sense of responsibility" less often than their Western counterparts. Instead, Asians place more emphasis on self-discipline and commitment, preciseness and meticulousness, resourcefulness and determination, and tenacity and steadiness [45]. Secondly, all our volunteers were college students, who tend to have relatively little contact with society and do not take much responsibility. Therefore, their understanding of self-consciousness and agreeableness may not be comprehensive, affecting the corresponding score on the self-esteem scale and further affecting the prediction performance of these two dimensions. Third, our research is based on facial images, in which there are obvious differences between the features of Chinese and Western people. For example, Westerners have obvious facial contours with high noses, while Asians have relatively flat facial contours and soft lines. Therefore, the prediction results of Chinese and foreigners, especially Westerners, personalities based on facial features are bound to be different. The above evidence illustrates the credibility of our results.

The ROC curves also showed that the model has good classification ability for neuroticism and extraversion but a slightly lower classification ability for openness, agreeableness, and conscientiousness.

Our research has shown that a person's personality has a certain relationship with his or her appearance. We estimated that machine learning (the deep learning network in our experiment) could reveal the multidimensional personality characteristics expressed based on the static shape of the face. We developed a neural network and trained it on a large dataset labeled with self-reported BF features without the participation of supervisory, third-party evaluators, avoiding the reliability limitations of human raters.

We further predicted that personality characteristics could be reflected in images of the entire face, not just in individual facial features. In order to verify our hypothesis, we developed S-NNPP, a deep neural network based on the attention mechanism, and added soft thresholding to achieve better prediction performance than existing

networks. Specifically, we compared its performance with that of the BP network and two kinds of high-performance deep neural networks, MobileNetV2 and ResNeSt50, and found that our S-NNPP network effectively had the best prediction accuracy (see Table 7). We identified three reasons for the improvement in the model accuracy. Firstly, we collected 13,347 pairs of data (including self-reported facial image and personality data), larger than any other dataset yet reported worldwide (the previous record was 12,447 pairs of data in [16]). Secondly, in terms of algorithm improvement, the excellent performance of ResNeSt in ImageNet image classification advantages of network in classification. We used the base module of ResNeSt to develop our improved network. Furthermore, although personality data are easily labeled as fuzzy, we improved the model's ability to process data containing different noises by introducing soft threshold techniques. Thirdly, in our dataset, the face images had relatively consistent backgrounds, distances to the camera, angles, and lighting, making later data processing more convenient.

4. Network Neural Network for Personality Prediction Based on 2.5D Images

In personality prediction, we found that some facial regions, such as the forehead, nose, cheekbones, and chin, whose features cannot be well located in the frontal face image; instead, profile images tend to be required for more accurate detection [23]. In fact, the facial information contained in the positive and lateral perspectives is naturally complementary. As a result, the combination of the two perspectives (i.e., 2.5D) is expected to reflect the relationship between facial images and personality more comprehensively than 2D images alone.

4.1. Experimental Setup. In this section, 360 students (180 males and 180 females) were selected from the previous 2D database for collecting additional facial images, including front and side images, as well as BF personality self-evaluation scores. We again used the 50% cross-validation method described previously for effect analysis; the data were scrambled and randomly divided into five parts, of which one part was further divided into equally sized test and validation sets; the other four parts served as the training set. Specifically, 288 images and the corresponding scores were used as the training dataset, and the remaining 72 images and corresponding scores were used as the test dataset.

It must be noted that the 2.5D images in the database included one front face image and two facial contour images. Experiments have shown that the geometric features of the left and right profiles are highly correlated, and most of the differences between the two sides can be described in the front face images; therefore, the side features were only extracted from the left profile images.

4.2. Results and Discussion. We tested the accuracy in predicting the five personality traits with different networks. Table 8 shows the F1 scores of the three deep learning networks for the five personality traits. Obviously, the frontal and lateral faces emphasize complementary facial regions; therefore, integration of the two kinds of image should result in more accurate personality measurements, motivating the use of 2.5D modeling used in this study. Following deep neural network training, the 2.5D prediction model achieved better personality prediction than the 2D model; particularly, the F1 score for extraversion increased to 93.02%, and the prediction performance for openness increased to 65.03%. This suggests that these two personality traits are more correlated with the information provided by facial contour images. There was no change, obvious or otherwise, in the prediction of other characteristics, which was directly related to the small size of the experimental 2.5D dataset.

Although we do not know how to train deep neural networks to learn human facial features, we know that the facial features extracted from the front and side images are completely different. Some facial features can be well described by front face images, while others can be accurately expressed in facial profile images. This indicates that the combination of the two perspectives provides more information about personality, so it is possible to further improve the performance of personality prediction.

5. Application

The personality characteristics of a person identified from facial images in real life can be used in many scenes. In daily social affairs, this technique is very useful for identifying personality types. Our method is a further development of traditional personality assessment methods. Facial-feature-based personality matching is expected to become a popular feature of all kinds of job hunting, social networking, and other similar websites, which can quickly recommend faces to users according to their preferences [45]. In addition, facial-feature-based personality prediction research can also be used in crime tracking [23], security inspections in transportation, driver evaluation, and target employee selection based on the images of faces in public security departments. We believe that considering the speed and low cost of this technology, its application potential is vast.

We recruited college students as the research objects in this study. In terms of application scenarios, our findings could help college students find jobs that suit their personalities in the form of "person-post matching"; the technology can provide a quick personality analysis to help employers interview employees when hiring. These models can also be applied to auxiliary functions such as student online consultation. However, we do not advocate solely relying on artificial intelligence to "identify people."

We conducted experimental analysis with an additional dataset to ensure the validity of our experimental results. This dataset consists of two groups of face images, each of which contains faceless photos of the corresponding sample that we initially collected. The two groups of pictures include images of students who qualified for postgraduate study without exams and images of students who were about to drop out due to failing a large number of courses or for violating school guidelines. Each group contains 8 face images, as shown in Figure 8.

Table 7: Prediction performance comparison.

Trait	Traditional BP (%)			MobileNetV2 (%)			ResNeSt50 (%)			S-NNPP (%)		
	TPR	FPR	F1	TPR	FPR	F1	TPR	FPR	F1	TPR	FPR	F1
Neuroticism	80.00	3.28	87.55	85.93	3.03	90.98	87.41	1.52	92.55	92.91	1.43	96.55
Extraversion	72.48	6.78	81.51	75.52	6.45	83.40	76.51	1.69	86.04	84.44	1.52	91.84
Openness	47.24	40.00	49.38	49.59	38.36	50.63	54.14	32.84	57.83	58.54	30.56	62.25
Agreeableness	44.59	45.45	50.54	55.12	35.71	56.68	55.81	34.78	57.83	60.50	32.43	63.25
Conscientiousness	41.29	50.00	46.55	54.42	33.33	59.93	58.39	30.77	62.26	60.69	26.23	65.42

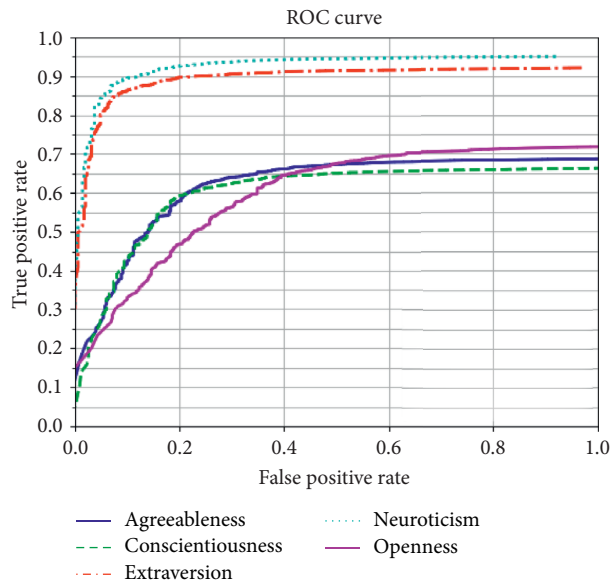
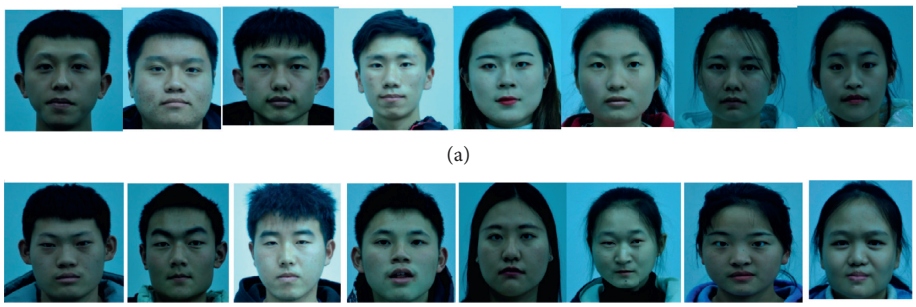


Figure 7: ROC curve.

Table 8: Prediction performance comparison for the 2.5D model.

Trait	MobileNetV2 (%)		ResNeSt50 (%)		S-NNPP (%)	
	F1 (2D)	F1 (2.5D)	F1 (2D)	F1 (2.5D)	F1 (2D)	F1 (2.5D)
Neuroticism	90.98	89.89	92.55	90.6	96.55	95.89
Extraversion	83.40	85.04	86.04	88.75	91.84	93.02
Openness	50.63	51.57	57.83	58.06	62.25	65.03
Agreeableness	56.68	55.34	57.83	56.45	63.25	62.87
Conscientiousness	59.93	57.06	62.26	60.09	65.42	65.10



(a)

(b)

Figure 8: Frontal images of two different types of students. (a) Eight students who have obtained the qualification of master's degree without examination. (b) Eight dropouts who failed many courses.

For these two types of samples, some of the subjects had excellent grades and strong scientific research capabilities, and some had failed subjects or dropouts due to rule violations; however, there was a general understanding of certain aspects of the personality traits of all subjects. In addition, because samples in the same category share certain characteristics and behaviors in common, there may be a certain degree of similarity between the personality traits of these individuals; therefore, we used the S-NNPP network to verify these results. We found that there are certain similarities in personality traits among samples of the same category. For example, for the 8 postgraduate candidates, the model predicted that conscientiousness and pleasantness were the most significant traits, while some also showed strong openness; for the students at risk of dropping out, their neuroticism was generally high, while no obvious commonality was observed in the other dimensions.

Data Availability

Our research involves a large number of face images. Before the photo collection, we signed a confidentiality agreement, which guarantees that the face images will not be disclosed to the public, so we cannot provide such data.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (61402371), the Shaanxi Provincial Science and Technology Innovation Project Plan (2013SZS15-K02), and the Shaanxi Provincial Key Scientific Research Project (2020zdlgy04-09).

Supplementary Materials

Supplementary material is the Big Five personality scale used in this paper, namely the NEO Personality Scale (the list of questions for self-personality assessment in this study). The 60-item scale measures the Big Five personality traits (Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism). (*Supplementary Materials*)

References

[1] E. L. Hicks, "On the characters of theophrastus." *The Journal of Hellenic Studies*, vol. 3, no. 11, pp. 128–143, 1882.

[2] D. C. Funder, "On the accuracy of personality judgment: a realistic approach," *Psychological Review*, vol. 102, no. 4, pp. 652–670, 1995.

[3] J. Luo and X. Dai, "Development of the Chinese adjectives scale of big-five factor personality III: based on the generalizability theory," *Chinese Journal of Clinical Psychology*, vol. 24, no. 1, pp. 88–94, 2016.

[4] D. J. Ozer and V. Benet-Martínez, "Personality and the prediction of consequential outcomes," *Annual Review of Psychology*, vol. 57, no. 1, pp. 401–421, 2006.

[5] N. N. Oosterhof and A. Todorov, "The functional basis of face evaluation," *Proceedings of the National Academy of Sciences*, vol. 105, no. 32, pp. 11087–11092, 2008.

[6] D. R. Carney, C. R. Colvin, and J. A. Hall, "A thin slice perspective on the accuracy of first impressions," *Journal of Research in Personality*, vol. 41, no. 5, pp. 1054–1072, 2007.

[7] K. Wolffhechel, J. Fagertun, U. P. Jacobsen et al., "Interpretation of appearance: the effect of facial features on first impressions and personality," *PLoS One*, vol. 9, no. 9, Article ID e107721, 2014.

[8] S. Alper, F. Bayrak, and O. Yilmaz, "All the dark triad and some of the big five traits are visible in the face," *Personality and Individual Differences*, vol. 168, Article ID 110350, 2021.

[9] M. Kosinski, "Facial width-to-height ratio does not predict self-reported behavioral tendencies," *Psychological Science*, vol. 28, no. 11, pp. 1675–1682, 2017.

[10] R. M. Godinho, P. Spikins, and P. O'Higgins, "Supraorbital morphology and social dynamics in human evolution," *Nature Ecology & Evolution*, vol. 2, no. 6, pp. 956–961, 2018.

[11] J. M. Carré and J. Archer, "Testosterone and human behavior: the role of individual and contextual variables," *Current Opinion in Psychology*, vol. 19, no. 2, pp. 149–153, 2018.

[12] R. Qin and W. Gao, "Modern physiognomy: an investigation o predicting personality traits and intelligence from the human face," *Science China Information Sciences*, vol. 61, Article ID 058105, 2018.

[13] R. Qin, *Personality Analysis Based on Facial Image*, University of Chinese Academy of Sciences, Beijing, China, 2016.

[14] E. Goeleven, R. De Raedt, L. Leyman, and B. Verschuere, "The karolinska directed emotional faces: a validation study," *Cognition & Emotion*, vol. 22, no. 6, pp. 1094–1118, 2008.

[15] P. V. Shebalin and W. Truszkowski, "Cooperation between intelligent information agents," in *Cooperative Information Agents V. CIA 2001*, M. Klusch and F. Zambonelli, Eds., vol. 2182, Springer, Berlin, Germany, 2001.

[16] A. Kachur, E. Osin, D. Davydov et al., "Assessing the big five personality traits using real-life static facial images," *Scientific Reports*, vol. 10, no. 1, 2020.

[17] Z. Zeng, *An Analysis of Students' Personality Traits Based on Face Features and Deep Learning*, Jiangxi Normal University, Nanchang, China, 2017.

[18] N. A. Moubayed, Y. Vazquez-Alvarez, A. McKay, and A. Vinciarelli, "Face-based automatic personality perception," in *Proceedings of the ACM Multimedia*, Orlando, FL, USA, November 2014.

[19] X. Wu and X. Zhang, "Automated inference on criminality using face images," 2017, http://arxiv.org/abs/1611.04135v2.

[20] S. Brahnam and L. Nanni, "Predicting trait impressions of faces using local face recognition techniques," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5086–5093, 2010.

[21] F. Gurpinar, H. Kaya, and A. A. Salah, "Combining deep facial and ambient features for first impressionestimation," in *Proceedings of the European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, 2016.

[22] C. Zhang, H. Zhang, S. X. Wei, and J. Wu, "Deep bimodal regression personality analysis," in *Proceedings of the ECCV Workshop Proceedings*, Springer, Amsterdam, The Netherlands, 2016.

[23] A. Laurentini and A. Bottino, "Computer analysis of face beauty: a survey," *Computer Vision and Image Understanding*, vol. 125, no. 8, pp. 184–199, 2014.

[24] T. Zhang, R.-Z. Qin, Q.-L. Dong, W. Gao, H.-R. Xu, and Z.-Y. Hu, "Physiognomy: personality traits prediction by

learning," *International Journal of Automation and Computing*, vol. 14, no. 4, pp. 386–395, 2017.

[25] A. Laurencin and A. Bettino, "Computer analysis of face beauty: a survey," *Computer Vision and Image Understanding*, vol. 125, no. 8, pp. 184–199, 2014.

[26] B. Rammstedt and O. P. John, "Measuring personality in one minute or less: a 10-item short version of the Big Five Inventory in English and German," *Journal of Research in Personality*, vol. 41, no. 1, pp. 203–212, 2007.

[27] G. Boyle and E. Helmes, "Methods of personality assessment," in *The Cambridge Handbook of Personality Psychology*, pp. 110–126, Cambridge University Press, Cambridge, UK, 2009.

[28] J. I. Biel, L. T. Mosquera, and D. G. Perez, "Facetube: predicting personality from facial expressions of emotion in online conversational video," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, Santa Monica, CA, USA, 2012.

[29] Z. Liu, Z. Jing, and W. Song, "From parzen window estimation to feature extraction: a new perspective," in *Proceedings of the Intelligent Data Engineering and Automated Learning–IDEAL 2018*, Madrid, Spain, November 2018.

[30] H. Esmaily, M. Tayefi, H. Doosti et al., "A comparison between decision tree and random forest in determining the risk factors associated with type 2 diabetes," *Journal of Research in Health Sciences*, vol. 18, no. 2, Article ID e00412, 2018.

[31] I. Rish, "An empirical study of the Naïve Bayes classifier," vol. 3, pp. 41–46, in *Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, vol. 3, pp. 41–46, IBM, New York, NY, USA, 2001.

[32] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] A. B. Khromov, *Te Five-Factor Questionnaire of Personality*, Kurgan State University, Kurgan, Russia, 2000.

[34] P. T. Costa and R. McCrae, "Domains and facets: hierarchical personality assessment using the revised NEO Personality Inventory," *Journal of Personality Assessment*, vol. 64, no. 1, pp. 21–50, 1995.

[35] R. R. McCrae and P. T. Costa, "A contemplated revision of the NEO five-factor inventory," *Personality and Individual Differences*, vol. 36, no. 3, pp. 587–596, 2004.

[36] O. John, E. Donahue, and R. Kentle, "The big five inventory-versions 4a and 54," Tech. Rep., Institute of Personality and Social Research, University of California, Berkeley, CA, USA, 1991.

[37] S. D. Gosling, P. J. Rentfrow, and W. B. Swann, "A very brief measure of the Big-Five personality domains," *Journal of Research in Personality*, vol. 37, no. 6, pp. 504–528, 2003.

[38] R. Q. Mario, M. David, T. Alexander et al., "Automatic prediction of facial trait judgments: appearance vs. structural models," *PLoS One*, vol. 6, no. 8, Article ID e23323, 2011.

[39] P. V. Shebalin, *Collective Learning and Cooperation between Intelligent Software Agents: A Study of Artificial Personality and Behavior in Autonomous Agents Playing the Infinitely Repeated Prisoner's Dilemma Game*, The George Washington University, Washington, DC, USA, 1997.

[40] R. Rosenthal, "Conducting judgment studies: some methodological issues," in *The New Handbook of Methods in Nonverbal Behavior Research*, Oxford University Press, Oxford, UK, 2005.

[41] X. Wu, X. Zhang, and C Liu, "Automated inference on sociopsychological impressions of attractive female faces," 2016, http://arxiv.org/abs/1611.04135v2.

[42] D. L. Donoho, "Denoising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 2002.

[43] S. Hu, J Xiong, P Fu et al., "Signatures of personality on dense 3D facial images," *Scientific Reports*, vol. 7, no. 1, p. 73, 2017.

[44] D. R. Valenzano, A. Mennucci, G. Tartarelli, and A. Cellerino, "Shape analysis of female facial attractiveness," *Vision Research*, vol. 46, no. 8-9, pp. 1282–1291, 2006.

[45] D. Wang and C. Hong, "Theoretical and empirical analysis on the differences between Chinese and western personality structure—taking Chinese personality scale (QZPS) and western five factor personality scale (NEO PI-R) as examples," *Acta Psychologica Sinica*, vol. 40, no. 3, pp. 327–338, 2008.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 1060182, 11 pages
https://doi.org/10.1155/2021/1060182

Research Article

ShipYOLO: An Enhanced Model for Ship Detection

Xu Han, Lining Zhao, Yue Ning, and Jingfeng Hu

Navigation College, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Lining Zhao; zhaolining@dlmu.edu.cn

Received 14 April 2021; Accepted 14 June 2021; Published 24 June 2021

Academic Editor: Chi-Hua Chen

The application of ship detection for assistant intelligent ship navigation has stringent requirements for the model's detection speed and accuracy. In response to this problem, this study uses an improved YOLO-V4 detection model (ShipYOLO) to detect ships. Compared to YOLO-V4, the model has three main improvements. Firstly, the backbone network (CSPDarknet) of YOLO-V4 is optimized. In the training process, the 3×3 convolution, 1×1 convolution, and identity parallel mode are used to replace the original feature extraction component (ResUnit) and more features are extracted. In the inference process, the branch parameters are combined to form a new backbone network named RCSPDarknet, which improves the inference speed of the model while improving the accuracy. Secondly, in order to solve the problem of missed detection of the small-scale ships, we designed a new amplified receptive field module named DSPP with dilated convolution and Max-Pooling, which improves the model's acquisition of small-scale ship spatial information and robustness of ship target space displacement. Finally, we use the attention mechanism and Resnet's shortcut idea to improve the feature pyramid structure (PAFPN) of YOLO-V4 and get a new feature pyramid structure named AtFPN. The structure effectively improves the model's feature extraction effect for ships of different scales and reduces the number of model parameters, further improving the model's inference speed and detection accuracy. In addition, we have created a ship dataset with a total of 2238 images, which is a single-category dataset. The experimental results show that ShipYOLO has the advantage of faster speed and higher accuracy even in different input sizes. Considering the input size of 320×320 on the PC equipped with NVIDIA 1080Ti GPU, the FPS and mAP@5 : 5:95 (mAP90) of ShipYOLO are increased by 23.7% and 13.6% (10.6%), respectively, with an input size of 320×320, ShipYOLO, compared to YOLO-V4.

1. Introduction

With the rapid development of deep learning in recent years, more and more deep learning techniques have been applied to intelligent ships [1, 2]. In 2020, Pan et al. proposed a fine-grained classification model RMA based on deep learning [3], which realizes the identification of navigation marks and provides accurate navigation mark information for intelligent ships. In 2021, Du et al. developed an intelligent navigation mark recognition system using deep learning technology [4], which provided an effective solution for intelligent ships. The vision system that uses computer vision technology to identify ships, navigation mark, and obstacles in the navigation environment has become an essential part of the intelligent ship perception system [5]. Therefore, an effective ship detection model is of great significance for improving the safety of intelligent ships.

There are many traditional object detection models proposed by researchers. Traditional object detection models mainly rely on region selection [6], feature extraction [7], and classifier classification [8]. In 2006, Dalal and Triggs proposed the HOG algorithm [9], which composes features by calculating and counting the histogram of the local area's gradient direction. Subsequently, Felzenszwalb et al. proposed the DPM algorithm [10], which produced corresponding excitation templates for image features and determined the target's location according to the distribution of excitation. However, object detection will predict many redundant borders. In response to this problem, Neubeck and Van Gool proposed the NMS algorithm [11] to eliminate redundant borders. This idea is also widely used in deep learning object detection models. Traditional object detection models have limitations in many aspects, and they cannot perform image features well. The rise of deep

learning in 2012 has had a massive impact on many fields, and object detection is no exception. A large number of the deep neural network parameters can extract features with better robustness and semantic relevance, and the performance of the classifier is also superior. Therefore, the object detection model based on deep learning can better learn the characteristics of the image. The object detection model based on deep learning mainly exists in two forms, two-stage and one-stage. The main difference is whether to predict the position information of the object's border and the border's category information in one step. In 2014, Girshick et al. used the idea of combining region candidates and CNN to propose a two-stage detection model R-CNN [12], which opened the chapter of deep learning for target detection. Based on R-CNN, Girshick proposed Fast R-CNN [13] to realize the end-to-end detection and convolution sharing function. In 2015, Ren et al. proposed the faster R-CNN [14] object detection model. The anchor frame idea and the region proposal network are designed, which significantly improves the R-CNN series of model' detection accuracy and won many firsts in the LSVRV and COCO competitions. In 2018, Redmon and Farhadi proposed YOLO-V3 [15], which added many excellent ideas to the network, such as residual ideas [16], multilayer feature maps [17], and no pooling layer. While ensuring the detection speed of the YOLO series, the detection accuracy of the model is improved. With the continuous improvement of deep learning technology, more and more methods are proposed to enhance object detection accuracy from different angles. In 2020, in order to improve the detection effect of analog instruments, Huang et al. proposed an improved YOLO-V3 algorithm in the robot-based detection process [18], which can effectively locate the instrument and has a good detection effect. In 2020, based on the original YOLO-V3, Bochkovskiy et al. integrated the excellent optimization strategies in the CNN field in recent years, including data processing, backbone network, network training, activation function, and loss function, and proposed a better one-stage object detection model YOLO-V4 [19]. Compared with YOLO-V3, the YOLO-V4 model uses a richer data enhancement method, including Mosaic data enhancement and SAT self-antagonism training. On the basis of the backbone network of YOLO-V3, the Mish activation function and the idea of CSPNet are introduced to increase the feature extraction effect of the backbone network. The SPP module is added behind the backbone network to further increase the receptive field of the model and further improve the detection effect.

Similarly, many ship detection models based on deep learning have been proposed by researchers. Like general object detection models, the ship detection model also has two-stage and one-stage forms. Li et al. proposed a SAR image ship detection model based on improved faster R-CNN [20]. As a two-stage detector, although the original detection accuracy is improved, the proposal filtering and ROI pooling operations limit the speed of the model, and it is difficult to achieve real-time detection. Wang et al. studied the application of SSD object detection model in ship detection under complex background [21] and used transfer learning technology to improve detection accuracy and overall performance. However, the single feature extraction network and FPN structure did not fully consider the small-scale ship's detection. Chen et al. used the attention mechanism to propose an improved YOLO-V3 (ImYOLO-V3) [22], and embedding the attention module into YOLO-V3 effectively improved the accuracy of detection, but there is no further optimization of the speed of the model. Jie et al. introduced the K-means clustering algorithm and soft nonmaximum suppression algorithm to optimize YOLO-V3 to make it more suitable for the ship scene [23], but the improved method proposed by it belongs to the engineering tuning technology, and there is no solution to the accuracy problem of ship detection from the perspective of model construction. Shan et al. combined camera and inertial sensor data and proposed a new marine target detection algorithm based on camera motion posture [24]. This algorithm uses the ideas of area candidate and edge detection to optimize the detection algorithm and improve the accuracy of ship detection. However, the traditional image enhancement method is still used, and its detection rate does not meet the requirements of the actual scene of the intelligent ship. In 2020, Li et al. proposed an improved ship detection algorithm LSDM based on YOLO-V3 and Densnet [25], which reduced the model parameters to 1/3 of the original YOLO-V3, but its backbone network uses a large number of densely connected structures. This design still affects the inference speed of the model.

In summary, the current ship detection models still has the problems of poor detection speed and missed detection of small-scale ships. First of all, in order to improve the detection speed and make the ship detection model achieve real-time effects, and this paper optimizes the backbone network of YOLO-V4. While ensuring the accuracy, the parameters of the model are reduced, and the inference speed of the model is effectively improved. Secondly, in order to solve the problem of missed detection of small-scale ships, this paper designs a new amplified receptive field module and combines the attention mechanism to optimize the original feature pyramid of YOLO-V4, which effectively improves the detection effect of small-scale ships. In the end, we get ShipYOLO, a faster and more accurate model for ship detection.

2. Methods

The YOLO-V4 model consists of a backbone network (CSPDarknet53), a receptive field amplification module (SPP), a feature pyramid (PAFPN), and a detection head (YOLOhead) (see Figure 1). The backbone network (CSPDarknet53) uses the CSP module composed of ResUnit components as the feature extraction part of the overall structure. The receptive field amplification module (SPP) uses pooling layers of different sizes to fuse features of different scales to amplify the receptive field. The Feature Pyramid Module (PAFPN) refers to PANet and obtains a two-layer pyramid structure. Although YOLO-V4 has good detection results overall, it has not been effectively designed for ship detection, so this paper has made targeted improvements to YOLO-V4.

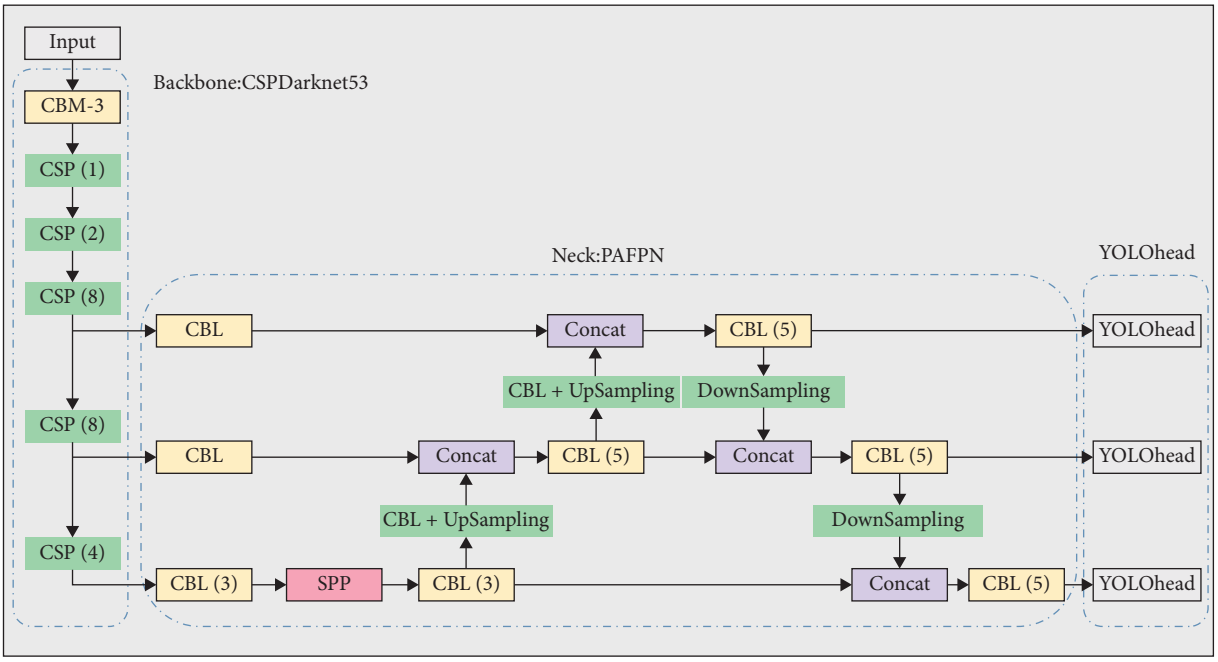


Figure 1: Schematic diagram of YOLO-V4 model structure.

2.1. Backbone Network Based on Structured Reparameterization (RCSPDarknet). The original ResUnit component of YOLO-V4 [16] is a typical multibranch structure (see in Figure 2(a)), and CBM_N is composed of N × N convolution, batch normalization, and activation function (Mish) in series. The calculation formula of the ResUnit component is shown as

$$y = F_1(F_2(x)) + x. \tag{1}$$

Although the multibranch topology has a good feature extraction effect, each branch's results need to be retained until superimposed or connected, which significantly increases memory usage and seriously affects the model's inference speed. Such a structure is very unfriendly to the ship detection field with high inference speed requirements. Therefore, removing the branch structure in the model can effectively improve the inference speed of the model. For example, the classic single-line model VGG [26], composed of multiple 3 × 3 convolution, although it has obvious advantages in speed, the accuracy is far inferior to the ResNet structure. Therefore, this study refers to the idea of RepVgg [27] and uses the structure reparameterization technology to construct the feature extraction component RepUnit (see in Figures 2(b) and 2(c)). Although the multibranch structure has poor inference speed, this structure is more conducive to model training and feature extraction. Therefore, in order to achieve both speed and accuracy improvements, this paper first uses a multibranch structure for training the calculation formula is as follows:

$$y = F_1(x) + F_2(x) + x. \tag{2}$$

Then, use the structure reparameterization technology to fuse the model parameters and convert a training block into a single 3 × 3 convolution layer for inference. The final calculation formula in the inference stage is shown as

$$y = F(x). \tag{3}$$

While ensuring the accuracy of the model, it effectively improves the inference speed of the model.

The structure reparameterization process and the calculation process of the convolution kernel are shown in Figure 3. First, the convolution layer and the batch normalization layer in the residual block are fused (this operation is performed in the inference stage of many deep learning frameworks), and the calculation formula is

$$\begin{aligned} W_i' &= \frac{\gamma_i}{\sigma_i} W_i, \\ \beta_i' &= -\frac{\mu_i \gamma_i}{\sigma_i} + \beta_i, \end{aligned} \tag{4}$$

where W_i is the convolutional layer parameters before calculation, β_i is the convolutional layer bias before convolution, μ_i is the mean value of the batch normalization layer, and σ_i is the variance of the batch normalization layer.

Branch (a) directly executes the fusion of the 3 × 3 convolution layer and the batch normalization layer, Branch (b) executes the fusion of the 1 × 1 convolution layer and the batch normalization layer, Branch (c) first sets a 3 × 3 convolution layer with a weight of 1 and then executes the fusion of the 3 × 3 convolution layer and the batch normalization layer (because this branch does not change the value of the input feature map, it is set to a 3 × 3 convolution layer with a weight value of 1, and then, it will maintain the original value after multiplying with the input feature map). Then, convert the convolution layer after branch (b) fusion into a 3 × 3 convolution layer (the value in the 1 × 1 convolution kernel is used as the center point of the 3 × 3 convolution kernel, and the other places are filled with 1). Finally, the 3 × 3 convolution layer in each branch are

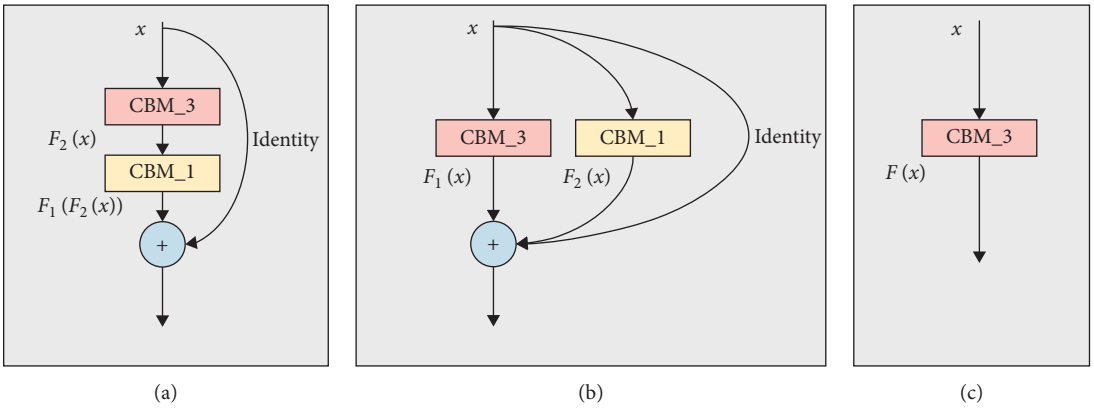


Figure 2: (a) ResUnit component of YOLO-V4. (b) RepUnit component of ShipYOLO training. (c) RepUnit component of ShipYOLO inference.

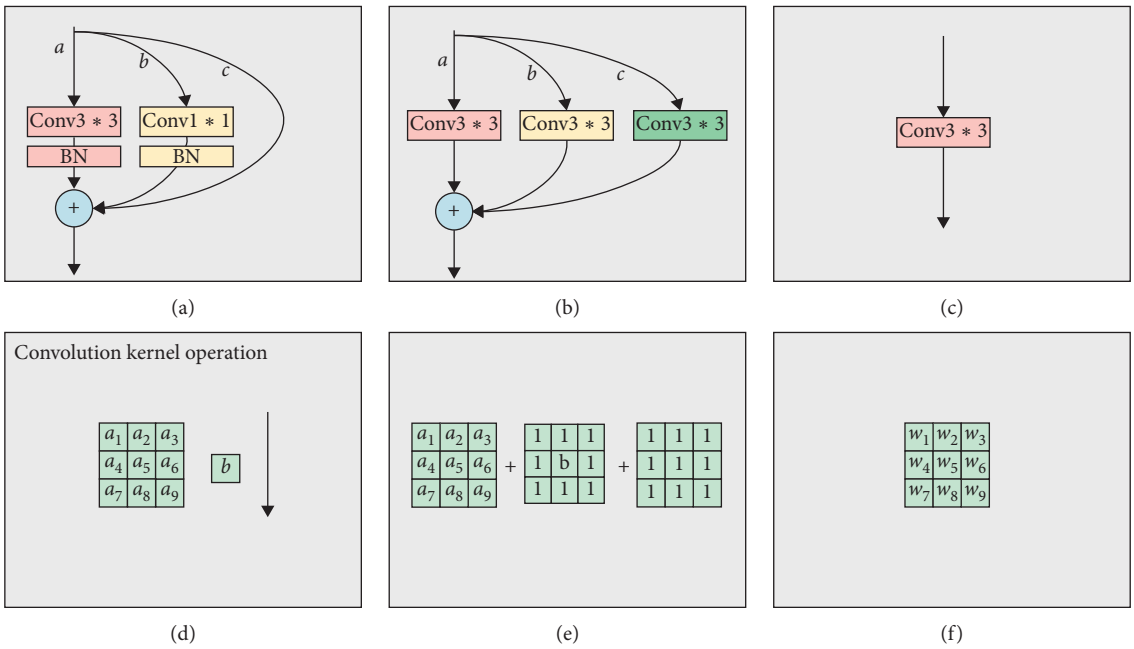


Figure 3: Structure reparameterization process of RepUnit component.

merged, and the weights and biases of all the branches are superimposed to obtain a 3×3 network layer after fusion.

In the end, we used the improved feature extraction component (RepUnit) to form a new module (RCSP) and got a new backbone network (RCSPDarknet), which effectively improved the model's inference speed and had a better detection effect.

2.2. Spatial Pyramid Pooling Module Based on Dilated Convolution (DSPP). YOLO-V4 was inspired by SPPNet [28] and added the SPP module (see in Figure 4(b)), CBL_N is composed of $N \times N$ convolution, batch normalization, and activation function (Leaky) in series (the difference from CBM_N is that they use different activation functions. In CBM_N, the activation function uses Mish and CBL_N uses Leaky),and MaxPool_N is the Max-Pooling layer whose kernel size is equal to N. The pooling operation of fixed blocks is used to stitch together different feature maps to realize the fusion of features of different sizes, which effectively improves the detection effect of images with significant differences in target size and increases the receptive field. However, ship sizes are different for ship detection, and the problem of missed small-scale ships is serious. The original SPP structure and traditional convolution structure are difficult to increase the receptive field while capturing small-scale targets in space. Luo et al. studied the problem of receptive fields in deep convolution networks [29] and pointed out that pixels in the center of the receptive fields are greater. In the forward pass process, the center pixel has more paths to transmit the pixel information to the neural node, and the edge pixels have fewer paths to transmit its pixel information to the neural node. Similarly, in the backward pass process, the receptive field's center pixel

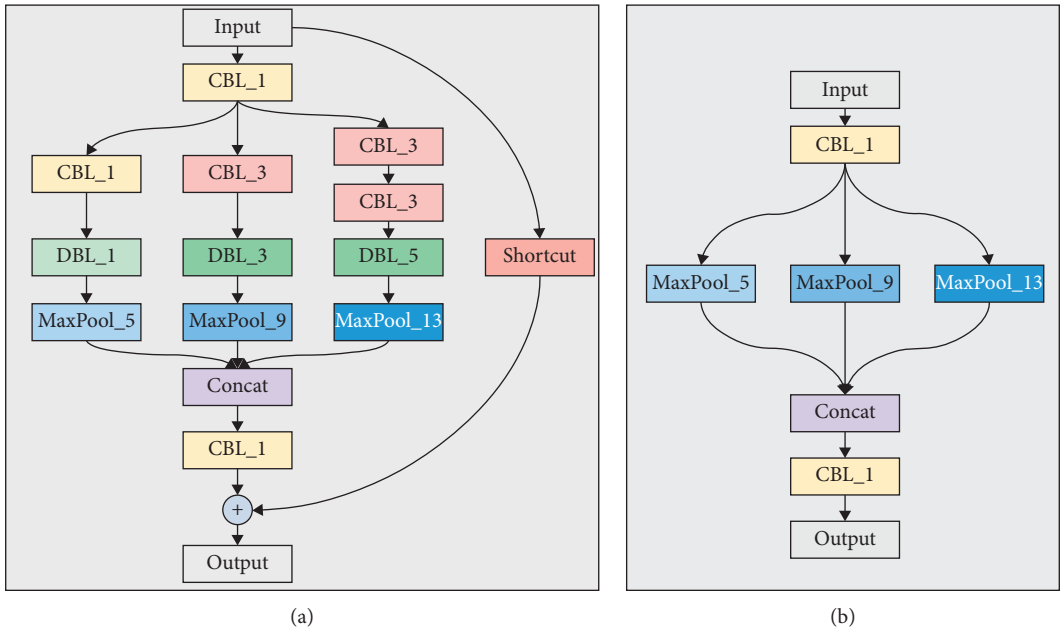


Figure 4: (a) DSPP structure in ShipYOLO. (b) SPP structure in YOLO-V4.

obtains more gradients from the corresponding neural nodes. The design of dilated convolution [30] can reduce the loss of spatial features without reducing the receptive field compared with ordinary convolution and can effectively consider the feature extraction of targets of different scales. Therefore, this paper refers to the dilated convolution and SPPNet, which designs a new feature enhancement module (DSPP). While increasing the model's receptive field, it improves its feature extraction effect for small-scale targets in space and effectively solves the problem of missing small-scale ships and improving ship detection accuracy. The DSPP structure is shown in Figure 4(a). DBL_N is composed of the dilated convolution with a spatial interval span of N, the batch normalization layer, and the activation function (Leaky) in series.

Firstly, the feature map is passed through a 1×1 convolution layer to reduce the number of channels and then divided into three branches. The three branches are composed of the Max-Pooling layer, convolution layer, and dilated convolution layer in series (the number of convolution kernels of each branch, the number of dilated convolution rates, and the kernel size of Max-Pooling are shown in Figure 4), and the last branch uses two 3×3 convolutions instead of 5×5 convolutions, reducing the parameters and deepening the nonlinear layer. Secondly, contact the feature maps of the three branches together and then connect to a 1×1 convolution layer which is used for the scale conversion feature. Finally, referring to the residual structure of ResNet, we get the feature enhancement module DSPP.

2.3. Feature Pyramid Based on Attention Mechanism (AtFPN). In deep learning, the fusion of different scales' features is an essential means to improve performance, and convolution layers learn semantic features of different levels of different depths. The FPN [31] structure proposed by Anthimopoulos simultaneously uses the high-resolution of low-level features and the high-semantic information of high-level features and achieves the prediction effect by fusing these features of different layers. YOLO-V4 effectively referred to this idea and combined with PANet [32] to add a bottom-up feature pyramid after the FPN layer to obtain PAFPN (see in Figure 1). This structure utilizes robust semantic features from the top to the bottom and strong positioning features from the bottom to the top and aggregates parameters from different backbone layers to different detection layers. However, in the PAFPN structure of YOLO-V4, the same convolution module as the backbone network is still used. Although it has a good feature extraction effect, it brings the problem of excessive parameter volume. Therefore, this paper refers to the CBAM structure, merges it with PAFPN, and adds a residual structure design to each semantic layer. We are obtaining a new feature pyramid structure (AtFPN), which effectively improves the model's accuracy and reduces the number of model parameters.

CBAM [33] was proposed by Woo et al. (see in Figure 5(a)). This structure provides attention maps from the channel and spatial dimensions, respectively, and is used for the middle feature map, which can effectively help the network's information. The channel attention module aims to focus on what features are meaningful. Firstly, the channel attention module compresses the spatial dimension of the input feature map, uses the Avg-Pooling layer and the Max-Pooling layer, obtains the global context information in the feature map while reducing the interference information in the feature map, then forwards it to a shared network (single-layer perceptron), and finally generates the channel attention map through sigmoid (see in Figure 5(b)). The calculation formula is shown as

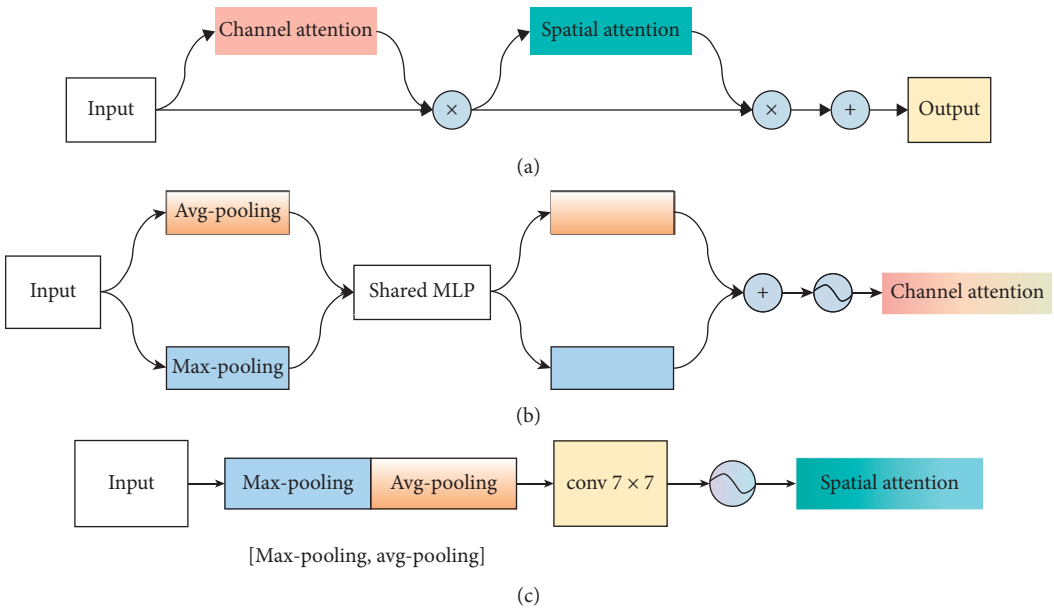


Figure 5: (a) Schematic diagram of the CBAM module structure. (b) Schematic diagram of the channel attention module structure. (c) Schematic diagram of the CBAM module structure.

$$y = \text{sigmoid}(\text{MLP}(\text{AvgPool}(x)) + \text{MLP}(\text{MaxPool}(x))). \tag{5}$$

Spatial attention is complementary to channel attention and aims to assign weights to feature maps to obtain spatially interesting areas (see in Figure 5(c)). Firstly, Avg-Pooling and Max-Pooling operations are applied along the channel axis, and they are connected to generate effective feature descriptors, and then, the spatial attention map is obtained through sigmoid. The calculation formula is as follows:

$$y = \text{sigmoid}(\text{Conv}(\text{concat}[\text{AugPool}(x); \text{MaxPool}(x)])). \tag{6}$$

Therefore, we replaced the original YOLO-V4 bottom-up semantic layer CBL component with a CBAM component, which effectively reduced the model parameters and increased the target region parameters' weight to be identified in the feature map at different scales. At the same time, we once again referred to the residual structure of ResNet in each semantic layer and fused the corresponding pixels of the shallow feature map output by the backbone network and the deep feature map after multilayer convolution to enhance the variety of feature map. The AtFPN designed in this paper is shown in Figure 6.

2.4. ShipYOLO. In summary, this paper designs ShipYOLO, a detection model that is more suitable for the ship field, and the model structure is shown in Figure 7. Firstly, an efficient backbone network RCSPDarknet is designed using the structure reparameterization technology, which effectively solves the current problem of low real-time performance in ship detection. Secondly, the feature enhancement module DSPP is designed using dilated convolution and Max-Pooling and combined with the feature pyramid structure AtFPN based on the attention mechanism, while ensuring the model inference speed, and it further improves the model accuracy and effectively solves the problem of small-scale ship missed inspection existing in the current ship detection model.

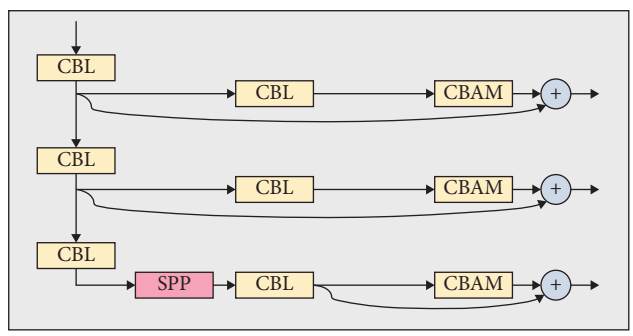


Figure 6: Schematic diagram of the AtFPN module structure.

3. Experiments and Results

3.1. Evaluating Indicator. This paper uses mAP as the model's accuracy evaluation indicator, where mAP@5 : 5:95 represents the average mAP at different IOU thresholds (from 0.05 to 0.95 and step size is 0.05). The mAP50 and mAP90 score tables represent mAP at IOU thresholds of 0.5 and 0.9. The mAP (small) represents the average mAP of small objects. FPS represents the number of frames transmitted per second. #Params represents the parameter amount of the model. For the convolutional layer, the calculation formula is shown as

$$\text{params} = C_o \times (k_w \times k_h \times C_i + 1), \tag{7}$$

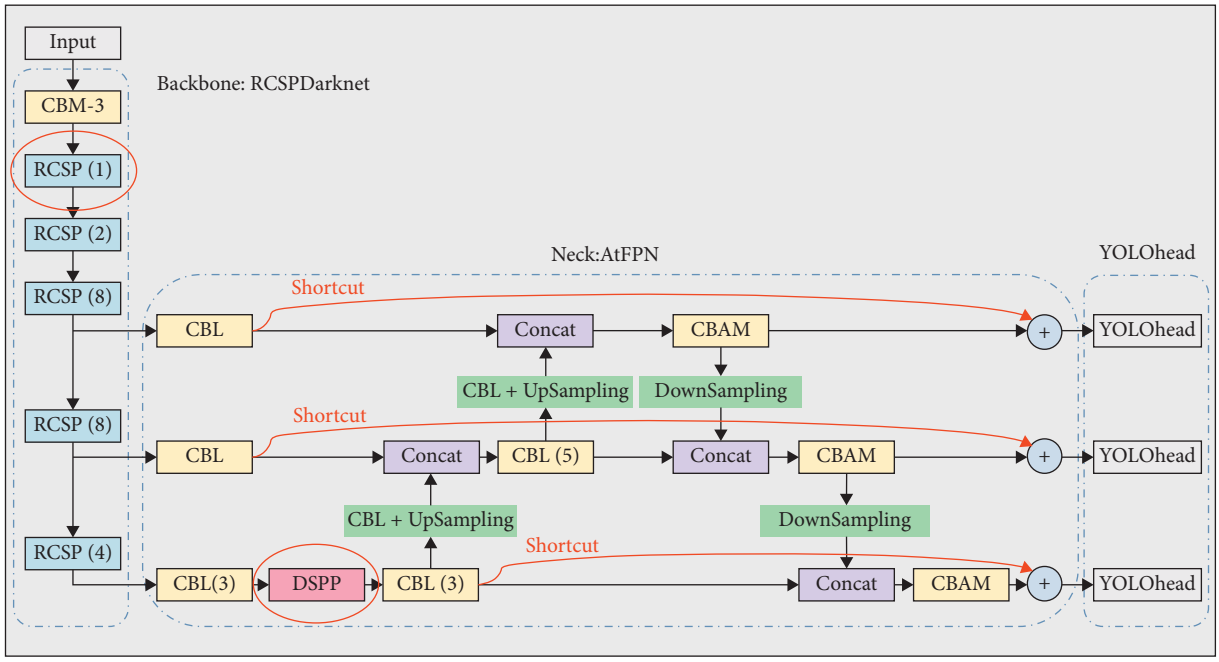


Figure 7: Schematic diagram of the ShipYOLO model structure.

where C_o is the number of output channels, C_i is the number of input channels, k_w is the width of the convolutional kernal, k_h is the length of convolutional kernal, and 1 is the bias of convolutional kernal.

For the fully connected layer, the calculation formula is shown as

$$\text{params} = (m + 1) \times n, \tag{8}$$

where m is the output vector dimension of the fully connected layer, n is the input vector dimension of the fully connected layer, and 1 is the bias of the fully connected layer.

3.2. Dataset

3.2.1. Public Ship Dataset. In 2018, Shao et al. created a public ship dataset (SeaShips) [34], which currently consists of 31,455 pictures, covering 6 common ship types (ore carrier, bulk cargo carrier, general cargo ship, container ship, fishing boat, and passenger ship). Part of the data is shown in Figure 8. For this dataset, we divided it according to the ratio of 8 : 2 and produced a training set and a validation set.

3.2.2. Self-Built Ship Dataset. In order to meet a richer scene, we have produced a ship dataset in a natural scene, a total of 2238 sheets, of which the category is a single category (Ship), and some of the dataset are shown in Figure 9. Similarly, we divided it according to the ratio of 8 : 2 and produced a training set and a validation set.

3.3. Experiment and Result. We conducted experiments in a 1080Ti environment. First, we used our three optimization strategies to conduct experiments on the basis of YOLO-V4. Using our self-built dataset with an input size of 512 × 512, the experimental results are shown in Table 1:

From Table 1, we can see that RCSPDarknet can significantly improve the inference speed of the model while maintaining the accuracy and reduce the amount of parameters. Embedding the DSPP module into YOLO-V4 can effectively improve the detection accuracy of the model, but the inference speed of the model is slower than that of YOLO-V4. Finally, the model parameters of YOLO-V4 based on AtFPN have been reduced, and the detection accuracy and inference speed have not been affected.

Finally, we compared the performance of ShipYOLO, YOLO-V4, and YOLO-V3 in the two datasets and tested the detection accuracy of three models for small targets. The experimental results are shown in Tables 2–4:

From Tables 2 and 3, we can see that YOLO-V4 has better detection accuracy than YOLO-V3 at input sizes of 416 × 416 and 512 × 512, while YOLO-V3 has better detection accuracy than YOLO-V4 at input sizes of 320 × 320, but YOLO-V4 has a faster inference speed. Through comparison and verification, the ship detection model ShipYOLO proposed in this paper is better than YOLO-V4 and YOLO-V3 in accuracy, FPS, and #Params. With an input size of 320 × 320, compared to YOLO-V4, ShipYOLO has a 13.6% increase in mAP@5 : 5:95 (10.6% mAP90), a 23.7% increase in FPS, and the model #Params reduced to 188 m. From Table 4, we can also find that our ShipYOLO has a better detection effect in the detection of small-scale ships.

We also screened some typical pictures for verification. As shown in Figures 10 and 11, it can be seen that ShipYOLO has solved the small-scale ship missed inspection problem of YOLO-V4 and YOLO-V3 and has a better bounding box regression effect. The comparison of Figures 10 and 11 and

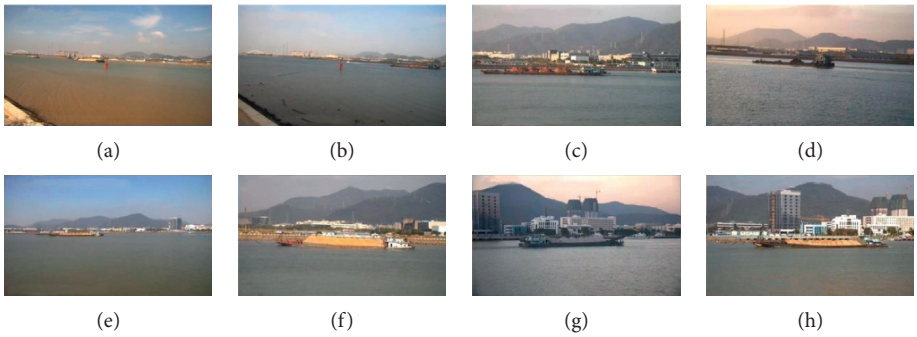


Figure 8: Dataset by seaships.

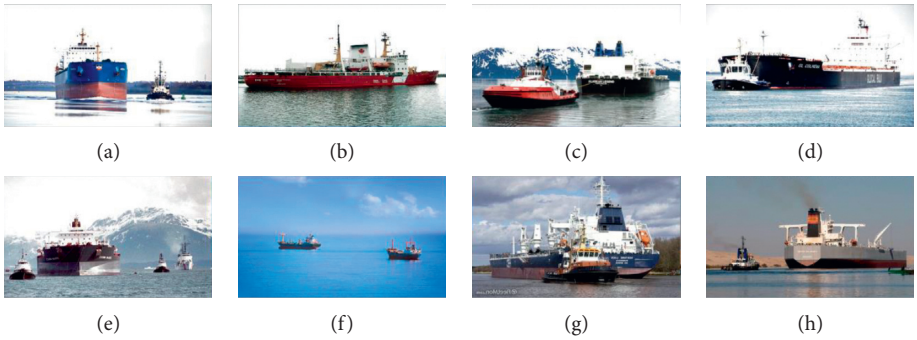


Figure 9: Dataset by ourselves.

Table 1: Experimental results of different optimization strategies.

Model	mAP90 (%)	FPS	#Params (MB)
YOLO-V4	89.4	38.3	244
YOLO-V4 + RCSPDarknet	88.1	52.6	220
YOLO-V4 + DSPP	91.3	34.8	255
YOLO-V4 + AtFPN	88.4	38.6	201

Table 2: Experimental results of seaships.

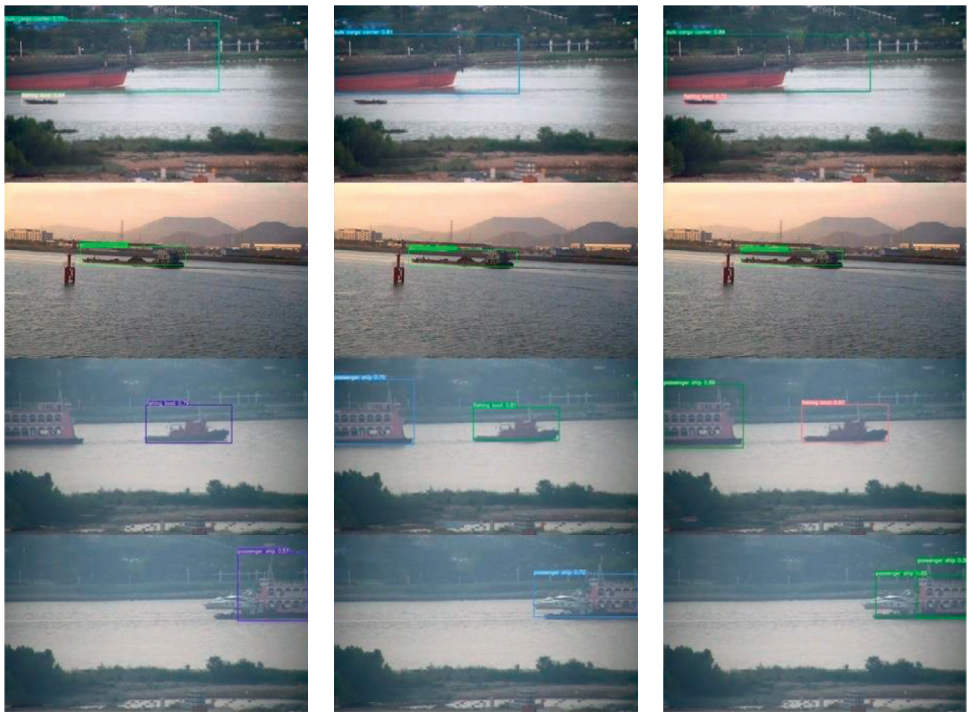
Model	InputSize	mAP@5 : 5 : 95 (%)	mAP50 (%)	mAP90 (%)	FPS	#Params (MB)
ShipYOLO	320	57.7	90.8	69.1	69.4	188
YOLO-V4	320	56.7	89.1	64.6	56.1	244
YOLO-V3	320	42.1	86.0	48.0	55.5	235
ShipYOLO	416	63.1	95.5	71.7	57.4	188
YOLO-V4	416	61.3	94.6	72.4	46.5	244
YOLO-V3	416	50.5	93.2	56.7	45.9	235
ShipYOLO	512	64.9	95.2	91.2	47.6	188
YOLO-V4	512	64.6	95.5	86.3	38.3	244
YOLO-V3	512	59.8	93.8	79.5	38.1	235

Table 3: Experimental results of the dataset by ourselves.

Model	InputSize	mAP@5 : 5 : 95 (%)	mAP50 (%)	mAP90 (%)	FPS	#Params (MB)
ShipYOLO	320	76.4	96.5	94.6	69.4	188
YOLO-V4	320	67.2	96.1	85.5	56.1	244
YOLO-V3	320	74.9	96.4	92.8	55.5	235
ShipYOLO	416	76.6	97.5	95.8	57.4	188
YOLO-V4	416	72.0	97.5	91.4	46.5	244
YOLO-V3	416	75.1	97.7	92.5	45.9	235
ShipYOLO	512	76.9	97.7	96.3	47.6	188
YOLO-V4	512	74.2	97.9	89.4	38.3	244
YOLO-V3	512	73.7	97.1	92.0	38.1	235

Table 4: Experimental results of the mAP (small).

	ShipYOLO	YOLO-V4	YOLO-V3
mAP (small)	70.0%	50.0%	35.0%
InputSize	512	512	512



Yolo-V3 (a) Yolo-V4 (b) ShipYOLO (c)

Figure 10: Validation result of the seaships.

Yolo-V3 (a) Yolo-V4 (b) ShipYOLO (c)

Figure 11: Validation result of the dataset by ourselves.

the experimental data in Tables 1 and 2 proves the effectiveness of ShipYOLO in the field of ship detection. It is a faster and more accurate ship detection model.

4. Conclusions

This paper proposes an enhanced model based on YOLO-V4 for ship detection. First of all, this paper uses structured reparameterization technology to optimize the backbone network. The new backbone network increases the model's inference speed, and effectively solves the problem of poor ship detection model speed. Secondly, this paper redesigns the amplified receptive field module of YOLO-V4 and optimizes the feature pyramid structure based on the attention mechanism. These structures improve the model's detection effect for small-scale ships and solve the problem of missed inspection of small-scale ships. Extensive experimental results show that our detection model ShipYOLO has a significant improvement in speed and accuracy compared to the current advanced detection models and can be effectively applied to the field of ship detection. Through experiments, we have also found that extreme weather conditions such as foggy weather and rainy days during the ship's navigation seriously affect the model's recognition of the ship. Therefore, we will also do more research in this section so that ships can be effectively identified in more complex environments.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 3132019400.

References

[1] Y. Tang and N. Shao, "Design and research of integrated information platform for smart ship," in *Proceedings of the International Conference on Transportation Information and Safety (ICTIS)*, pp. 37–41, Banff, AB, Canada, August 2017.

[2] A. García-Domínguez, "Mobile applications, cloud and big-data on ships and shore stations for increased safety on marine traffic; a smart ship project," in *Proceedings of the International Conference on Industrial Technology (ICIT)*, pp. 1532–1537, Seville, Spain, March 2015.

[3] M. Pan, Y. Liu, J. Cao, Y. Li, C. Li, and C.-H. Chen, "Visual recognition based on deep learning for navigation mark classification," *IEEE Access*, vol. 8, pp. 32767–32775, 2020.

[4] Y. Du, S. Sun, S. Qiu et al., "Intelligent recognition system based on contour accentuation for navigation marks," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6631074, 11 pages, 2021.

[5] I. Im, D. Shin, and J. Jeong, "Components for smart autonomous ship architecture based on intelligent information technology," *Procedia Computer Science*, vol. 134, pp. 91–98, 2018.

[6] J. Chen, D. Q. Chen, and S. H. Meng, "A novel region selection algorithm for auto-focusing method based on depth from focus," in *Proceedings of the Euro-China Conference on Intelligent Data Analysis and Applications*, pp. 101–108, Xian, China, October 2017.

[7] A. Tiwari, A. Kumar, and G. M. Saraswat, "Feature extraction for object recognition and image classification," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 10, 2013.

[8] J. C. Platt, "Sequential minimal optimization: a fast algorithm for training support vector machines," *Support Vector Machines*, Technical report, Microsoft, Redmond, WA, USA, 1998.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, San Diego, CA, USA, June 2005.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester et al., "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[11] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 3, pp. 850–855, Hong Kong, China, August 2006.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[13] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[15] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, http://arxiv.org/abs/1804.02767.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[17] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[18] J. Huang, J. Wang, Y. Tan, D. Wu, and Y. Cao, "An automatic analog instrument reading system using computer vision and inspection robot," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6322–6335, 2020.

[19] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, http://arxiv.org/abs/2004.10934.

[20] J. Li, C. Qu, and J. Shao, "Ship detection in SAR images based on an improved faster R-CNN," in *Proceedings of the 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, pp. 1–6, Beijing, China, November 2017.

[21] Y. Wang, C. Wang, and H. Zhang, "Combining a single shot multibox detector with transfer learning for ship detection using sentinel-1 SAR images," *Remote Sensing Letters*, vol. 9, no. 8, pp. 780–788, 2018.

[22] L. Chen, W. Shi, and D. Deng, "Improved YOLOv3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images," *Remote Sensing*, vol. 13, no. 4, p. 660, 2021.

[23] Y. Jie, L. Leonidas, F. Mumtaz, and M. Ali, "Ship detection and tracking in inland waterways using improved YOLOv3 and deep SORT," *Symmetry*, vol. 13, no. 2, p. 308, 2021.

[24] X. Shan, D. Zhao, M. Pan, D. Wang, and L. Zhao, "Sea-sky line and its nearby ships detection based on the motion attitude of visible light sensors," *Sensors*, vol. 19, no. 18, p. 4004, 2019.

[25] Z. Li, L. Zhao, X. Han, and M. Pan, "Lightweight ship detection methods based on YOLOv3 and DenseNet," *Mathematical Problems in Engineering*, vol. 2020, Article ID 4813183, 10 pages, 2020.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[27] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: making VGG-style ConvNets great again," 2021, http://arxiv.org/abs/2101.03697.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[29] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," 2017, http://arxiv.org/abs/1701.04128.

[30] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 472–480, Honolulu, HI, USA, July 2017.

[31] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, and S. G. Mougiakakou, "Semantic segmentation of pathological lung tissue with dilated fully convolutional networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 714–722, 2018.

[32] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018, http://arxiv.org/abs/1803.01534.

[33] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," 2018, http://arxiv.org/abs/1807.06521.

[34] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: a large-scale precisely annotated dataset for ship detection," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4413505, 13 pages
https://doi.org/10.1155/2021/4413505

Research Article

A Fuzzy Logic-Based Approach for Humanized Driver Modelling

Yuxiang Feng,[1] **Pejman Iravani**,[2] **and Chris Brace**[3]

[1]*Department of Civil and Environmental Engineering, Imperial College London, London, UK*
[2]*Department of Mechanical Engineering, University of Bath, Bath, UK*
[3]*The Institute for Advanced Automotive Propulsion Systems, University of Bath, Bath, UK*

Correspondence should be addressed to Yuxiang Feng; y.feng19@imperial.ac.uk

Received 11 April 2021; Accepted 16 June 2021; Published 23 June 2021

Academic Editor: Chi-Hua Chen

All human drivers can be characterised by their habitual choice of driving behaviours, which results in a wide range of observed driving patterns and manoeuvres. Developing control strategies for autonomous vehicles that address this feature would increase the public acceptance of such vehicles. Therefore, this paper proposes a novel approach to developing rule-based fuzzy logic driver models that simulate different driving styles in the car-following regimes. These driver models were trained with the collected on-road driving data to capture corresponding human drivers' characteristics. The proposed approach consists of three main components: collecting on-road driving data, developing a vehicle model, and establishing the car-following driver models. Firstly, an instrumented vehicle was used to collect driving data over the same route for three consecutive months. Car-following scenarios during these journeys were extracted, and related data were processed accordingly. Afterwards, a representative model of the instrumented vehicle was created and evaluated. Finally, a fuzzy logic driver model that uses humanized inputs was developed and calibrated with the recorded data. The developed driver model's performance was assessed using the collected driving data and a baseline PID driver model. With the performance validated, models representing more aggressive and more defensive driving styles were derived following the same procedure. A cross-driver analysis was then implemented in a normalized car-following scenario with the established vehicle model to investigate the impacts of different driving styles further. The developed driver model can introduce driving styles into drive cycle experiments and replicate on-road real driving emission tests in the laboratory. Moreover, as the proposed method has high robustness to incomplete datasets, it can be a more cost-effective option to facilitate the development of humanized and customized vehicle control strategies for autonomous driving.

1. Introduction

Initially proposed by Elander et al. [1], the concept of driving style, or vaguely referred to as driver behaviour occasionally, can be defined as "driving style which concerns the way individuals choose to drive or driving habits that have become established over a period of years." While the consensus of a unified definition about this concept has not yet been reached [2–5], it generally refers to the driver's own habitual choice of driving manoeuvre, which can be reflected in both cognition and action characteristics of human drivers. The cognition aspect consists of the driver's evaluation of the current traffic scenario and the corresponding decision-making process. Meanwhile, the action aspect mainly relates to the driver's adopted driving patterns, such as the preference on the pattern of accelerating, braking, and gear shifting.

Similar to the vagueness in the definition of driving style, its exact classification can also vary significantly with the focus of research. For example, it can be as simple as two groups (aggressive and normal) [6, 7] or categorized explicitly into eight groups (angry, high-velocity, risky, anxious, dissociative, patient, careful, and distress-reduction) [8]. Nonetheless, it should be noted that the most common trend in existing studies is to classify it into three distinct groups, namely, aggressive, normal, and defensive.

Aggressive driving style, or referred to as sporty, hostile, and angry driving style, is the behavioural pattern consisting of risky speeding, abrupt speed change, harsh acceleration and deceleration, and improper lateral position maintenance

[9]. This type of driving style has received the most extensive research focus since it deviates from the norm and expected behaviour of a driver and can be a significant cause of increased fuel consumption and emissions and potentially even deadly crashes [7, 10]. The second group of driving style, as indicated by its name, refers to the driving style that is most commonly witnessed which is neither too aggressive nor too defensive. Normal driving style often functions as a reference to isolate other driving styles, which can be used as the baseline for driving style classification. Defensive driving is often conceptualised as contrary to aggressive driving [11]. While it has not been specifically defined, defensive driving generally refers to moderate acceleration/deceleration, properly maintained headway distance, and careful traffic flow participation. It has a high correlation with normal driving but in a more passive manner.

Various studies have been conducted to facilitate the classification of driving styles. For example, methods based on fuzzy logic [12–14] and neural network [6, 15] were developed to directly differentiate driving styles from driving data. Other supervised [16] and unsupervised models [17, 18] have also been developed to facilitate driving style classification. Moreover, instead of using manoeuvre frequencies, Li et al. [19] categorized highway driving behaviours into 12 manoeuvre states and focused on the transition patterns with a random forest algorithm. It was found that a better driving style estimation can be achieved using the transition probabilities between manoeuvres.

While extensive studies have been conducted to investigate the influence of driving style on driving safety [20–22] and fuel consumption [23–25], they were mainly based on analysing human driving data collected under various scenarios. Although positive correlations between driving style and each factor were found within these studies, the exact influence of driving style difference is still under debate, especially for fuel consumption research. This is mainly caused by the unrepeatable nature of human behaviours, making comparative study challenging to implement. A viable solution is to develop a driver model that can mimic human behaviours and perform different driving styles to solve this issue.

Although the driving style difference can be reflected in various driving scenarios, such as car-following, free flow, and driving under instructions, car-following scenarios are preferred in this study for investigation. This is because car-following regimes are the most primary scenario encountered by human drivers nowadays [26]. Additionally, both cognition and action characteristics of driving style can be reflected in the driver's preferred headway distance and driving patterns in this regime. Therefore, this study aims to develop humanized driver models specifically in car-following regimes.

The major contribution of this paper is three-folded. First, a fuzzy logic-based humanized driver modelling approach was developed, which is capable of simulating different driving styles using environmental inputs from human driver perception. Second, an anchored procedure to the standard World-wide harmonized Light duty Test Cycle (WLTC) was proposed to facilitate the driver models' evaluation. Third, the difference in driving style and their influence on fuel consumption was evaluated in a unified comparative environment.

The remaining of this paper is structured into four sections. Section 2 briefly reviews the relevant literature, and Section 3 explains the adopted methodology. Afterwards, the obtained results and discussion of analysis are presented in Section 4. Finally, Section 5 summaries the findings and limitations of this study and suggests future studies' direction.

2. Literature Review

With the earliest research implemented back in the 1950s [27], various approaches have been adopted to develop car-following models, which can be roughly divided into two categories, namely, the explanatory and nonexplanatory models [28].

Explanatory models are mainly based on closed-form mathematical models and usually define an ego vehicle's movement as a function of headway distance, relative speed, and host speed. Existing studies adopting this approach include Tang et al. [29]. They incorporated individual preference on optimal speed and safe distance into the traditional full velocity difference (FVD) model as additional driver attributions. Another study based on FVD was implemented by Zhang et al. [30], which considered the acceleration of the preceding vehicle and the ego driver's driving style. Both studies have divided driving styles into three categories (aggressive, neutral, and conservative) and extracted features of each category to establish the respective driver models. Both studies found the developed models can effectively improve stability and realism than the conventional FVD models.

Model predictive control is another commonly adopted approach for developing explanatory models. For instance, Luo et al. [31] developed an MPC-based ACC algorithm that considered safety, comfort, fuel consumption, and car-following performance as additional constraints. Meanwhile, Zhang and Vahidi [32] investigated the adoption of eco-driving strategies and incorporated the prediction of the front vehicle's behaviour into their model, leading to the improved capability to effectively reduce fuel consumption.

These explanatory models' significant advantage is their computation efficiency, which enables them to be easily incorporated in traffic flow simulation, especially with a large scale of vehicles. Consequently, these models have been widely adopted in microscopic and macroscopic traffic simulation studies [29, 30]. However, it should be noted that although the driving style parameters in some studies were calibrated from naturalistic driving data, these models' performance in mimicking human driving styles still lacks evaluation.

Unlike explanatory models defined using mathematical relations, nonexplanatory models are data driven, and the ego vehicle's movement is usually directly calibrated. Many data-driven models have been adopted in this research field. For example, Su et al. [33] categorized 84 drivers into three driving style groups and extracted features of each group to

calibrate three representative driver models based on fuzzy logic. Meanwhile, another study based on fuzzy logic was conducted by Hao et al. [34]. They developed two generalised driving style models (aggressive and conservative) using selected vehicle trajectory data from a published dataset. A genetic algorithm was adopted for the calibration of the fuzzy membership function.

Artificial neural network (ANN) is another widely adopted approach for human driver modelling due to its performance in imitating human behaviours from the machine learning perspective. The earliest research can be traced back to 1998, when Macadam et al. [6] developed a car-following driver model based on a two-layer neural network. Although only one driver model was calibrated, it has achieved satisfying results in the preliminary evaluation, revealing the promising potential of developing driver models using neural networks. Bifulco et al. [35] developed three driver models based on ANN, linear function, and polynomial function. These models were calibrated and evaluated using the same dataset. While no significant difference was found among the three models, the three-layer feedforward neural network model showed more flexibility in learning capabilities. Moreover, a series of studies were conducted by Shi et al. [36–38], which developed driver models using different variations of ANN, e.g., cerebellar model articulation controller [36], radial basis function [37], and wavelet neural network [38]. The developed networks receive vehicle speed and speed change as input and generate throttle position and brake pressure as output. The personalised models were then employed to follow the FTP-75 drive cycle to normalize the driving style variations. While no comparisons were made among the adopted neural networks, their capabilities of capturing the behavioural characteristics of those drivers were confirmed.

Some modern machine learning approach has also been adopted in similar research. For instance, Wang et al. [39] trained a recurrent neural network (RNN) with gated recurrent units (GRUs), which yielded higher simulation accuracies than traditional models that only use instantaneous inputs. Meanwhile, Gao et al. [40] used inverse reinforcement learning to establish each driver's reward function and analysed their driving characteristics and car-following strategies. Jiang et al. [41] used maximum likelihood inverse reinforcement learning to estimate driving style parameters from the collected driving data. A longitudinal assistance strategy was then developed with the calibrated parameters based on inverse reinforcement learning. Zhu et al. [42] developed a deep deterministic policy car-following gradient model, which uses the disparity between simulated and observed speed as the reward function to reproduce car-following behaviours.

It can be noted that all these nonexplanatory models attempted to infer the inner relations between human driver's cognition and action directly from the naturalistic driving data, collected from either on-road instrumented vehicles or driving simulators. One significant advantage of these approaches over exploratory models is the improved performance in simulating humanized behaviours, as they are directly inferred from driving data. While these data-driven models' benefits are apparent, they also suffer from some drawbacks, most notably the expensive computation requirements and difficulty in interpretation. Owing to these algorithms' relatively more complex architecture, they have higher requirements on the hardware platform. Moreover, large amounts of naturalistic driving data are also required by these models for training and inferring. Therefore, these data-driven models are more commonly used in simulating microscopic traffic scenarios with a limited number of vehicles.

While these existing studies have achieved some success in modelling human driving styles, it should be noted that they mainly take the accurate measurements of the ego vehicle's speed and the speed difference between ego and the preceding vehicle as input to their models. As human drivers' perception of the traffic environment is based on estimations, these accurate measurements can impair the ability to capture cognition characteristics. Moreover, these variables may be inappropriate from human drivers' perception. Therefore, this study proposes to develop a fuzzy logic-based modelling approach, which uses more humanized inputs and can incorporate the vagueness in drivers' perception to improve the similarity to human reasoning. Moreover, it should be noted that fuzzy logic models also are more computationally efficient than other data-based models and are more robust to incomplete data and less accurate traffic environment measurements.

3. Materials and Methods

3.1. Naturalistic Driving Data Collection. To facilitate the naturalistic driving data collection, a VW Sharan was instrumented to record daily driving on a selected route (shown in Figure 1) for three consecutive months. The chosen route covers approximately 45 miles and consists of both urban and rural segments. During the data collection phase, a sum of 90 trips was recorded, with an average duration of 63 minutes. The recorded maximum vehicle speed during each trip was approximately 100 km/h.

The vehicle state information was directly retrieved from the OBD-II port using an Influx Rebel CT OBD data logger. Meanwhile, a continental 77 GHz long-range ARS 308 radar and a Nextbase 512G dashcam were also instrumented on the vehicle to facilitate the collection of corresponding traffic information.

During the postdata processing, both the headway distance and the relative speed between the preceding and ego vehicles were extracted from the radar measurements. Simultaneously, the recorded dashcam footages were processed to generate a second source of headway distance measurements. Afterwards, a Kalman filter was developed to fuse these measurements and generate optimized headway distance estimation [43]. Synchronization between vehicle state information and traffic information was then implemented to form the naturalistic driving dataset for developing the driver model.

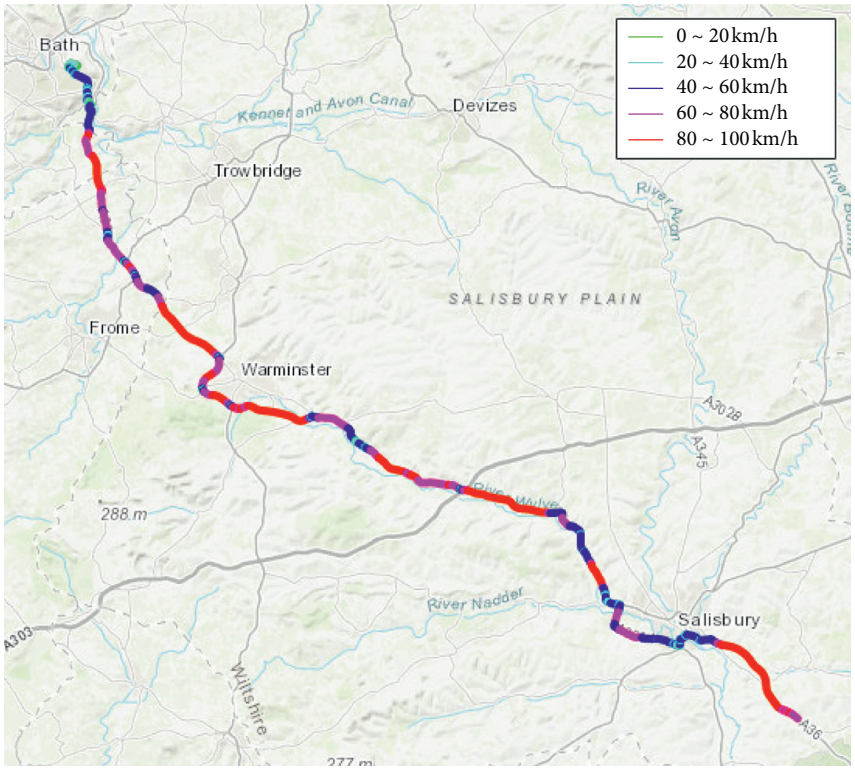


Figure 1: Route for driving data collection.

3.2. Vehicle Simulation Model. To facilitate the evaluation of the established driver model, a vehicle model that possesses the instrumented vehicle's essential features was developed. Using the vehicle parameters specified in [44], a corresponding vehicle model was developed in Simulink, a graphical programming environment for modelling, simulating, and analysing multidomain dynamical systems. It consists of five subsystems, namely, engine, brake, transmission, tire, and vehicle body. The architecture of the established vehicle model is illustrated in Figure 2.

3.3. Fuzzy Logic Driver Modelling. A fuzzy logic controller typically consists of five components: variables, rules, a fuzzifier, an inference engine, and a defuzzifier. Meanwhile, it also comprises four steps, initialization, fuzzification, inference, and defuzzification. Variables and rules are defined in the initialization phase. Variables are linguistic terms describing inputs and outputs, usually ranging from small to large. For each variable, it has a set of linguistic terms and associated membership functions.

Meanwhile, rules are the logic linkage between inputs and outputs and are used to determine correlating actions to each specific combination of inputs. Both parameters can be defined using the expert's knowledge during the initialization. After the initialization phase, the controller's actual inputs are transformed from crisp values to corresponding linguistic terms. Typically, two or more terms can be correlated with each input, and membership functions determine the degree of involvement. With the fuzzified inputs and corresponding rules, an inference engine can be used to compute each rule's contribution to the output. Finally, the accumulated output is transformed from fuzzy into crisp during the defuzzification phase.

With the general procedure of fuzzy logic outlined, it can be noted that fuzzy rules are the kernel of fuzzy logic and have a vast influence on its performance. Thus, these rules need to be appropriately defined to improve controller performance. To define such rules, the model's inputs and outputs need to be properly selected in advance. From drivers' perception perspective, multiple inputs can be received during driving, such as headway distance, relative speed to the preceding vehicle, current vehicle speed, engine rpm, gear selection, pedal position, road gradient, time, and weather. While all this information can influence drivers' decisions, it is unrealistic to consider all of them together, as the number of fuzzy rules will increase dramatically with multiple inputs. Thus, only the most prominent parameters, such as headway distance, current vehicle speed, gear selection, and pedal position, were selected to simplify the rule composition.

Meanwhile, two sets of headway distances were derived to improve the similarity to human drivers. While the actual distance measurement was used in a short range, the time gap measurement was adopted in middle and long ranges. This is mainly because of the widespread adoption of the two-second time gap rule in most countries. Drivers commonly obey it to assess safety headway distance by counting two seconds using stationary references. While this rule is more likely a reaction time-based guidance, it is useful when vehicle travels above a certain speed. A

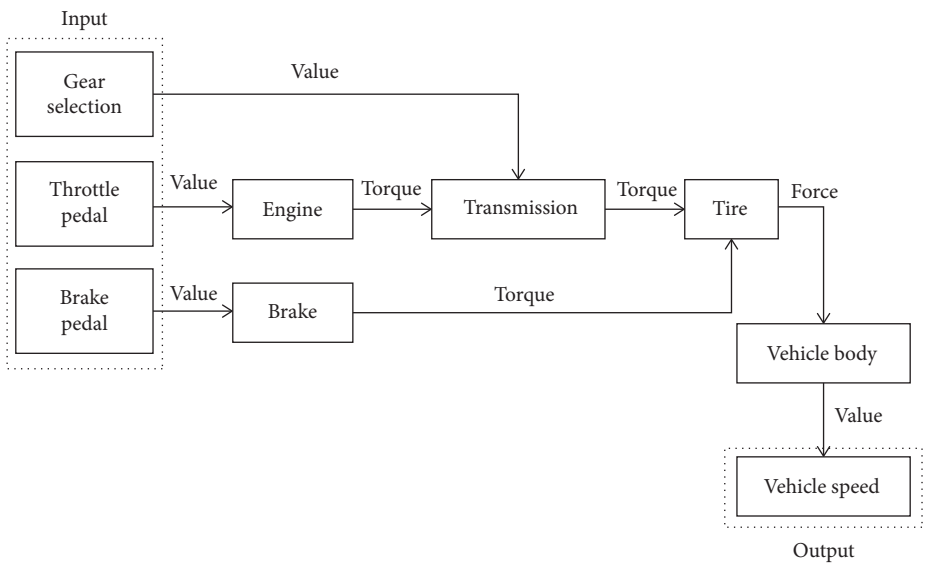


Figure 2: Architecture of the established vehicle model.

physical distance to a distant preceding vehicle can be challenging to estimate. However, this rule does not apply to slow-moving vehicles, and physical distance can be easily estimated instead. Therefore, both time and distance-based headway distances were used segmentally as model inputs, together with current vehicle speed, gear selection, and pedal position.

The typical driver outputs can be denoted as throttle pedal movement, brake pedal movement, and gear shifting. As throttle and brake pedals cannot be activated simultaneously and multiple outputs can increase fuzzy rules' complexity, power demand was selected as the sole output to simplify the fuzzy logic controller. A linear transformation was created between power demand and pedal movements. Thus, pedal commands can be derived accordingly from the output of the fuzzy logic controller. A gear shifting strategy based on vehicle speed was also developed.

According to experts' knowledge from the corresponding participant, five linguistic terms were derived for time gap (Very Small, Small, Middle, Large, and Very Large), current vehicle speed (Very Slow, Slow, Appropriate, Fast, and Very Fast), and power demand (Harsh Decelerate, Slight Decelerate, Maintain, Slight Accelerate, and Harsh Accelerate), respectively, and three for distance measurement (Close, Medium, and Far). Meanwhile, based on the number of variables, 15 rules were derived for the distance-based model and 25 rules for the time gap model. The control surface of each rule set is illustrated in Figure 3. The created rules share a similar structure, and one example can be described as follows: IF headway_distance is Close AND vehicle_speed is Fast THEN power_demand is Harsh Decelerate.

Meanwhile, Triangular-shaped membership functions were selected for each variable, which can be denoted as

$$f(x) = \left\{ \begin{array}{ll} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0 & c \le x \end{array} \right\}, \tag{1}$$

where a, b, and c are thresholds used to define the shape and location of the membership function.

As these thresholds have a vital influence on the established fuzzy controller's performance, they need to be correctly determined to optimize the controller. Therefore, these parameters should be calibrated using collected naturalistic driving data. There were 15 parameters associated with its membership function for a variable containing five linguistic terms and 9 for a variable with three linguistic terms. Car-following scenarios were isolated from the collected three months real driving data to calibrate these parameters. The recorded headway distance data and vehicle state information were synthesized and paired with the driver's intention, denoted by pedal movements and gear selection.

The method proposed by Yadav and Yadav [45] was adopted for the calibration of these parameters. According to their theory, each category of data was first sorted in ascending order, and then, K means clustering was performed to cluster these values into separate groups. The number of clusters was determined by the quantity of linguistic terms for each variable. Meanwhile, the cluster centre value was assigned to b, the central vertex parameter. Moreover, the similarity value between adjacent data were computed as

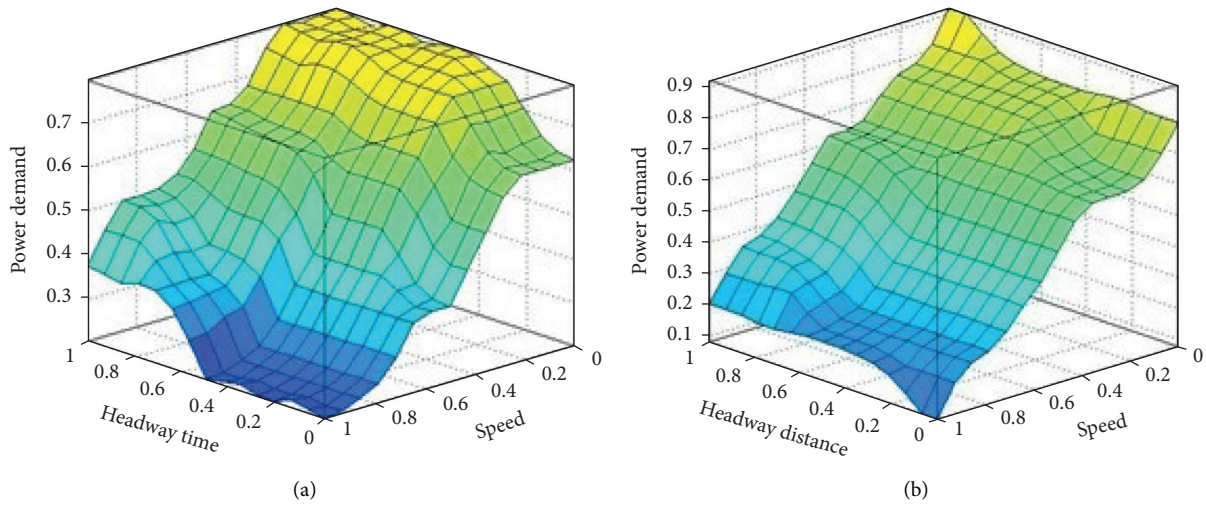


Figure 3: Control surface for each rule set. (a) Time based. (b) Distance based.

$$S = \begin{cases} 1 - \dfrac{v_{i+1} - v_i}{C \times \sigma_s}, & \text{if } v_{i+1} - v_i \leq C \times \sigma_s, \\ 0, & \text{else}, \end{cases} \tag{2}$$

where S is the similarity, v_i is the ith data, C is the control parameter, and σ_s is the standard derivation of $v_{i+1} - v_i$.

Afterwards, each cluster's minimum value of similarity was selected as the membership value of two boundary points for that cluster. Therefore, the remaining two defining parameters can hence be computed as

$$a = b - \frac{b - y_{\min}}{1 - S_{\min}}, \tag{3}$$

$$c = b + \frac{y_{\max} - b}{1 - S_{\min}}. \tag{4}$$

The obtained membership functions for each variable are illustrated in Figure 4.

To improve the simulation speed, this driver model was programmed as an s-function block. This setting can effectively reduce the consumed compiling time of the entire simulation model. Moreover, it can be noted that, for the two headway measurements, the test driver shows wider cognition of the short range, which indicates the tendency of maintaining medium or long headway distance. Meanwhile, the driver also favours driving at medium speed and has a wider recognition of very fast speed. As for the power demand, slight decelerate and slight accelerate cover the widest cognition zone. Thus, the driver prefers to change vehicle speed mildly and generally avoids harsh acceleration and deceleration. This calibration result indicated that the test driver possessed a normal to defensive driving style, coinciding with the driver's self-assessment and classification results [17].

3.4. Simulation Scenario. To facilitate the evaluation of the driver model's difference, a simulation scenario was developed to unify the comparison environment based on

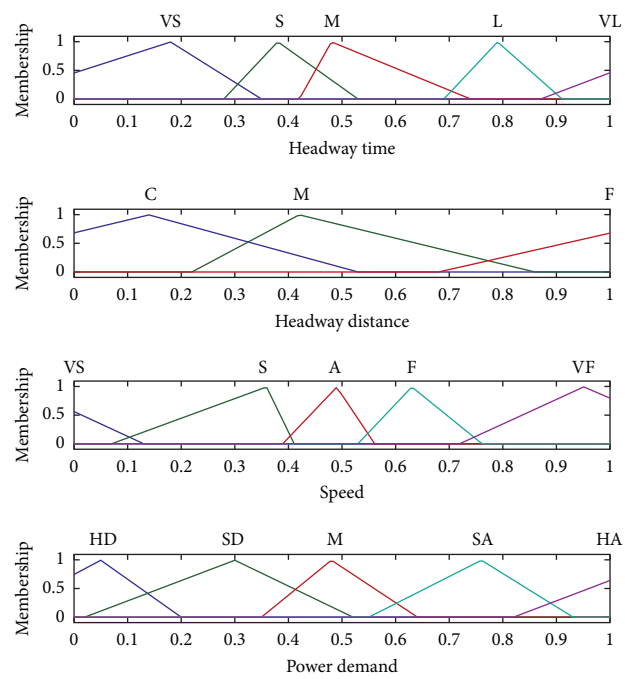


Figure 4: Calibrated membership functions.

WLTC, the latest drive cycle for emission test. Based on the power-weight ratio of the adopted instrumented vehicle, the Class 3 WLTC test cycle was selected [46], as illustrated in Figure 5.

To introduce driving style into this drive cycle data, one possible solution is assuming the driver is instructed to follow this speed profile. The driving style variance can then be partially reflected in the oscillations, convergent speed, and tracking accuracy. While these parameters can reveal the influence of driving style to a certain extent, they are not directly correlated, especially from the driver's perception. Therefore, an anchored procedure was proposed to convert this speed profile into a car-following scenario to incorporate the cognition characteristic of human drivers. A

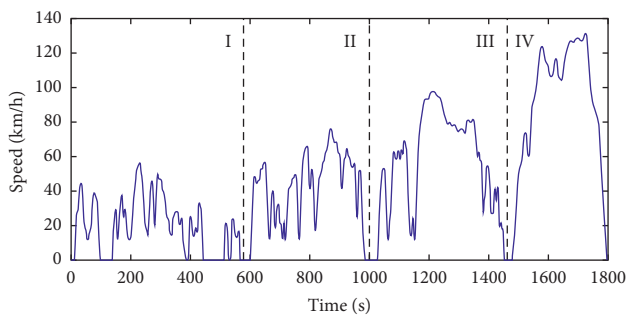


Figure 5: WLTC class 3 speed profile.

simulated preceding vehicle is introduced to create such a simulation scenario, which performs the WLTC speed profile. Thus, instead of directly providing the drive cycle data to the driver model, the drive cycle information is converted as the headway distance variations between the model and the simulated preceding vehicle. Hence, this setting can allow the driving style variances recorded in real-world car-following scenarios to be reflected in procedures that are anchored to standard drive cycle tests. While this setting can increase the difference between actual and target speed profiles, the similarity to human drivers is improved, and the influence of driving style can be more directly revealed.

Meanwhile, as shown in Figure 5, it can be noted that the WLTC drive cycle can be divided into four parts, namely, Low, Medium, High, and Extra High. Owing to the selected route, the majority of collected naturalistic driving data was within Low to High speed segments, which indicated that the calibrated driver model was also tuned explicitly for this speed range. Thus, the Extra High speed segment was discarded. Therefore, only the first three segments (0–1460s) of the WLTC drive cycle were used in this study.

4. Results and Discussion

This section consists of four sections. The established vehicle model is evaluated first. Afterwards, an obtained driver model was compared with the corresponding human driver to examine the proposed modelling approach's performance. Moreover, two more driver models were developed to facilitate the investigation into the characteristics of different driving styles and their influence on fuel consumption. Finally, the limitation and direction for future work of this study is discussed.

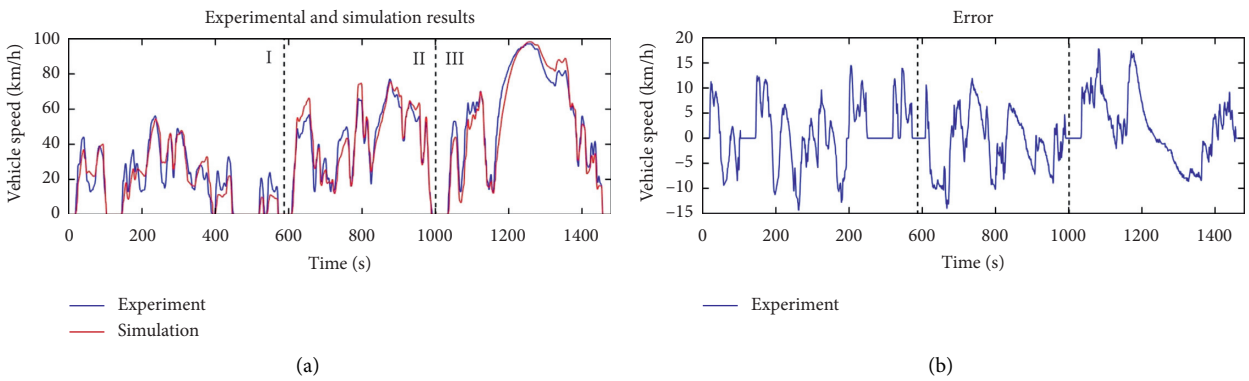
4.1. Vehicle Model Validation. Before examining the driver model's performance, the established VW Sharan vehicle model should be validated as a simulation basis. As this study's overarching aim is to evaluate the driver model using the WLTC drive cycle, it should be worthy to first assess the vehicle model's performance with recorded actual inputs from the corresponding vehicle. Therefore, the 4WD AVL RoadSimTM 48″ Chassis Dynamometer and the Stähle Autopilot SAP2000 robot driver within the Centre for Low Emission Vehicle Research of the University of Bath were used to acquire related experimental data. The Sharan was installed on the chassis dynamometer, and the robot driver was instructed to follow the WLTC Class 3 drive cycle speed profile. The vehicle state and robot driver information were recorded using a Rebel data logger. Essential information, such as vehicle speed, throttle pedal position, gear selection, engine rpm, robot accelerator leg, and brake leg movement, was extracted. While throttle pedal position, brake pedal position, and gear selection were required as vehicle model inputs, brake pedal position could not be directly recorded by the data logger. Therefore, a mitigation solution of obtaining brake pedal position was created for this validation. As the accelerator leg and brake leg movements of the Stäle robot were also recorded and the relation between pedal positions and leg movements should be fixed after installation, a mapping between vehicle pedal and robot leg was created using the throttle pedal and the accelerator leg. Afterwards, the brake pedal position can hence be obtained with this established mapping and record brake leg movement information. Both pedal positions and gear selection were transmitted to the vehicle model as inputs. The obtained vehicle speed profile is shown in Figure 6, together with the corresponding experimental result and computed error.

As illustrated in Figure 6, it can be noted that the simulation results generally possessed all the basic features of experimental data. As listed in Table 1, the total duration, accumulated stop duration, maximum and average speed of simulation, and experiment matched well between simulation and experiment.

4.2. Driver Model Examination. With the vehicle model validated, the established driver model's performance can hence be examined using the proposed simulation scenario. The speed profile of the first three segments of the WLTC drive cycle was assigned to the preceding vehicle. The total simulation time was set to 1460s. Meanwhile, an initial gap of 2 m was also assigned between the preceding and ego vehicle. Moreover, to increase the similarity to human drivers and boost simulation speed, a pedal change delay of 0.5s was also introduced to the system to simulate human drivers' common cognitive delay time [47]. The calibrated normal driver model was examined first, and corresponding results are shown in Figure 7.

As shown in Figure 7(a), although the speed profile varies from the WLTC drive cycle, the basic features and tendencies are preserved. It can be noted that the driver model shows some oscillations, which were introduced by the 0.5s pedal change delay. Meanwhile, the headway distance data illustrated in Figure 7(b) show that the driver model could perform a proper car-following behaviour, with the mean and maximum headway distance to be 12.50 m and 33.42 m, respectively.

To evaluate the established fuzzy logic controller's performance in simulating human driving style, a traditional PID-based driver model was also developed and tested in the same simulation scenario. As the calibrated fuzzy logic driver model can be classified as a normal driver and

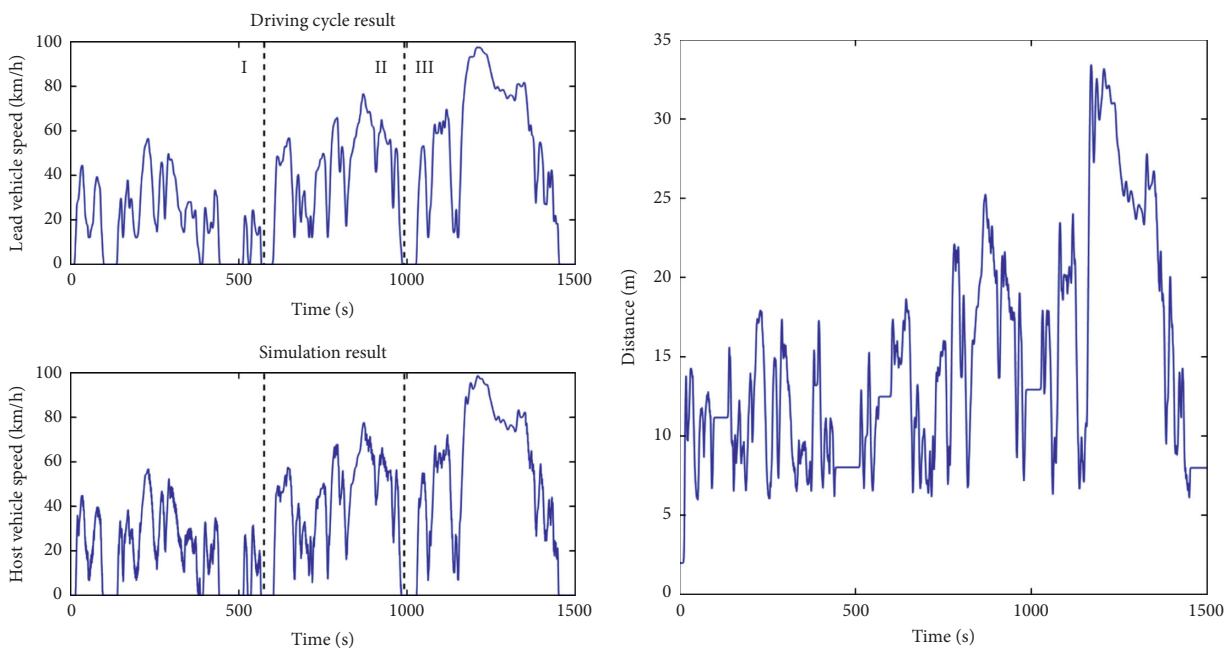


(a) (b)

Figure 6: Speed comparison between simulation and experiment.

Table 1: Comparison between simulation and experiment.

Category	Experiment			Simulation		
	I	II	III	I	II	III
Duration (s)	573	418	466	571	418	467
Stop duration (s)	140	35	42	145	37	45
Max speed (km/h)	56	77	97	54	76	97
Average speed (km/h)	19.2	40.1	60.0	23.1	39.4	57.2



(a) (b)

Figure 7: Simulation results of normal driver model. (a) Speed. (b) Headway distance.

generally maintain the 2s time gap, the PID driver model was designed to also hew to the same time gap and follow a leading vehicle performing WLTC drive cycle. As a direct measure of driver's characteristic, throttle pedal demand and time gap were selected to compare the human driver and two simulation models. Meanwhile, both variables for the human driver were averaged from all naturalistic driving data to reduce any potential basis in a single trip. The obtained probability density distribution of throttle pedal position and time gap are illustrated in Figure 8.

As shown in Figure 8(a), it can be noted that the variations of throttle pedal distribution between the human driver and fuzzy controller are much smaller than the difference between the human driver and PID controller. The PID controller possessed a higher proportion of throttle value between 70% and 100%, which indicated that

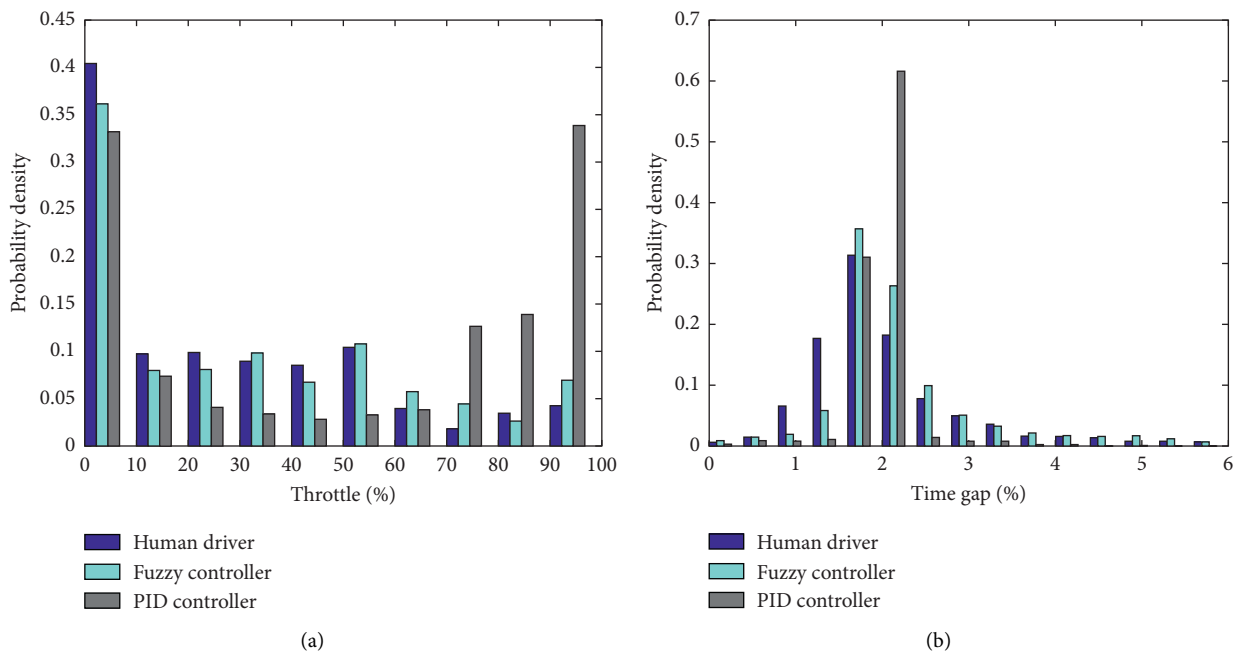


Figure 8: Driver model examination. (a) Throttle pedal. (b) Time gap.

the PID controller accelerated harsher to maintain the required time gap. Although the human driver had a larger proportion for the smaller throttle value, this may be mainly caused by complex traffic conditions and repeated stop-start scenarios. Aside from this segment, the probability distribution of the human driver and the fuzzy controller between 10% and 100% was similar, with the largest difference being 0.05, which occurred between 20% and 30%. From the distribution of probability density of throttle demands, it can be noted that the established fuzzy logic controller shows a better performance in simulating the human driver. Moreover, the time gap distribution was also extracted and illustrated in Figure 8(b). The intervals were increased to 0.4s for better visualization. It can be noted that the human driver shows a larger proportion of smaller time gaps, which is similar to the throttle pedal distribution and could be the consequence of complex traffic scenarios. Although there were some variations between the fuzzy controller and human driver's time gap distribution, the established fuzzy controller still outperformed the PID controller.

4.3. Driving Style Comparison. To facilitate the investigation into the difference of driving style, the same procedure was repeated to develop two additional driver models. Following the driving style classification results in [17], these two human participants possess a relatively more aggressive and defensive driving style, respectively. The obtained membership functions of these two driver models are illustrated in Figure 9.

According to Meiring and Myburgh [9], aggressive driving style, or referred to as sporty, hostile, or angry driving style occasionally, is a behavioural pattern containing risky speeding, abrupt speed change, harsh acceleration and deceleration, and improper lateral position maintenance. Meanwhile, defensive driving is defined as contrary to aggressive driving, which generally refers to moderate acceleration and deceleration, well-maintained headway distance, and careful participation in the traffic flow [48]. It has a high correlation with normal driving with a more passive manner. It can be noted from Figure 9 that the aggressive driver favours smaller safety distances and faster car-following speed, while the defensive driver possesses the opposite trend. For example, when headway time is 0.5, it will be perceived by the aggressive driver as a combination of "Middle" and "Large" and defensive driver as "Small" and "Middle," which reflects the cognitive difference of these driving styles. Both calibrated driver models were also examined in the proposed simulation scenario to evaluate these driving styles and investigate their influence on fuel consumption.

As shown in Figure 10(a), the headway distances vary among these three styles. It was found that the mean and maximum headway gaps were 6.76 m and 23.52 m for the aggressive model, 12.50 m and 33.42 m for the normal model, and 17.94 m and 52.46 m for the defensive model. Thus, it can be noted that the aggressive model showed a greater tendency of tailgating, while the defensive model remained the largest safe distance. Moreover, to further evaluate this tendency, the headway distance distributions of each driver model were computed. It was found that the aggressive model is more distributed (82.5%) in the small headway distance zone (0–10 m), while the defensive model occupies the largest proportion (76.9%) for a headway gap larger than 15 m. This revealed variation of headway distance is in coincidence with the common definitions of these three driving styles, as the headway gap is a crucial measure of aggressivity.

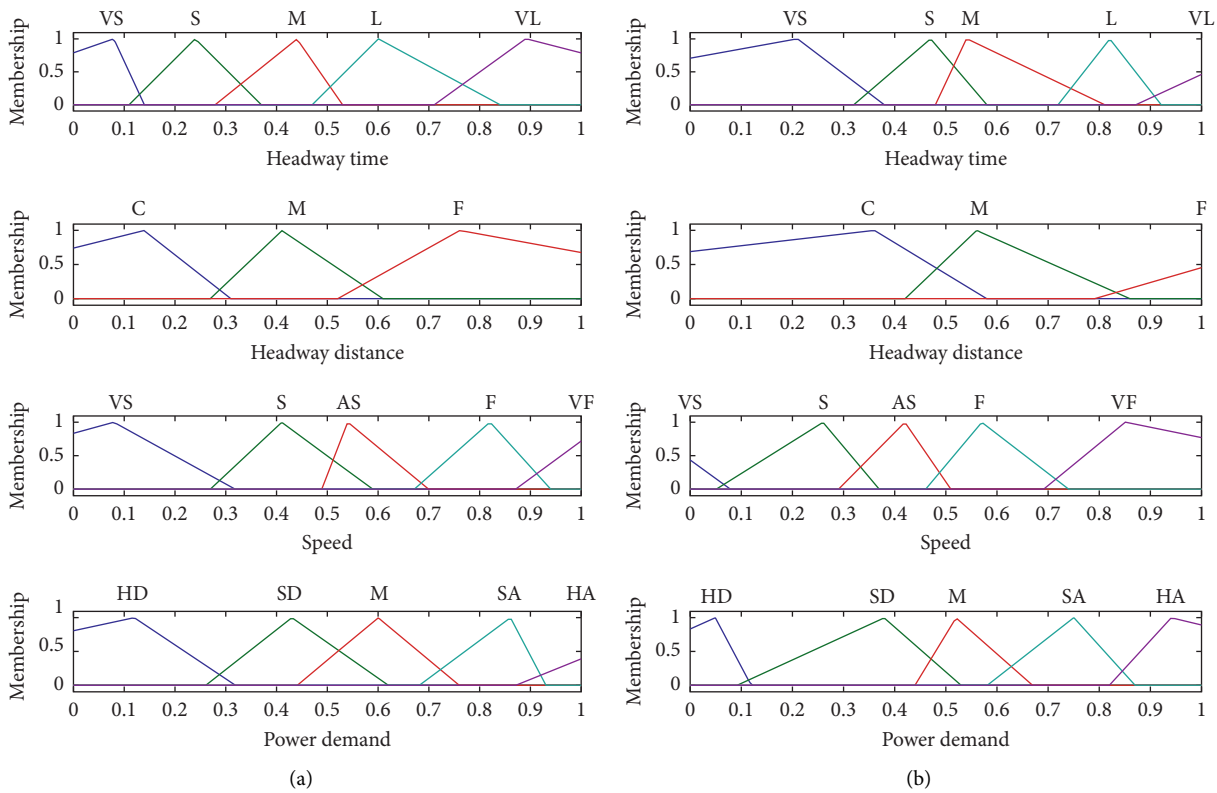


Figure 9: Calibrated membership functions. (a) Aggressive model. (b) Defensive model.

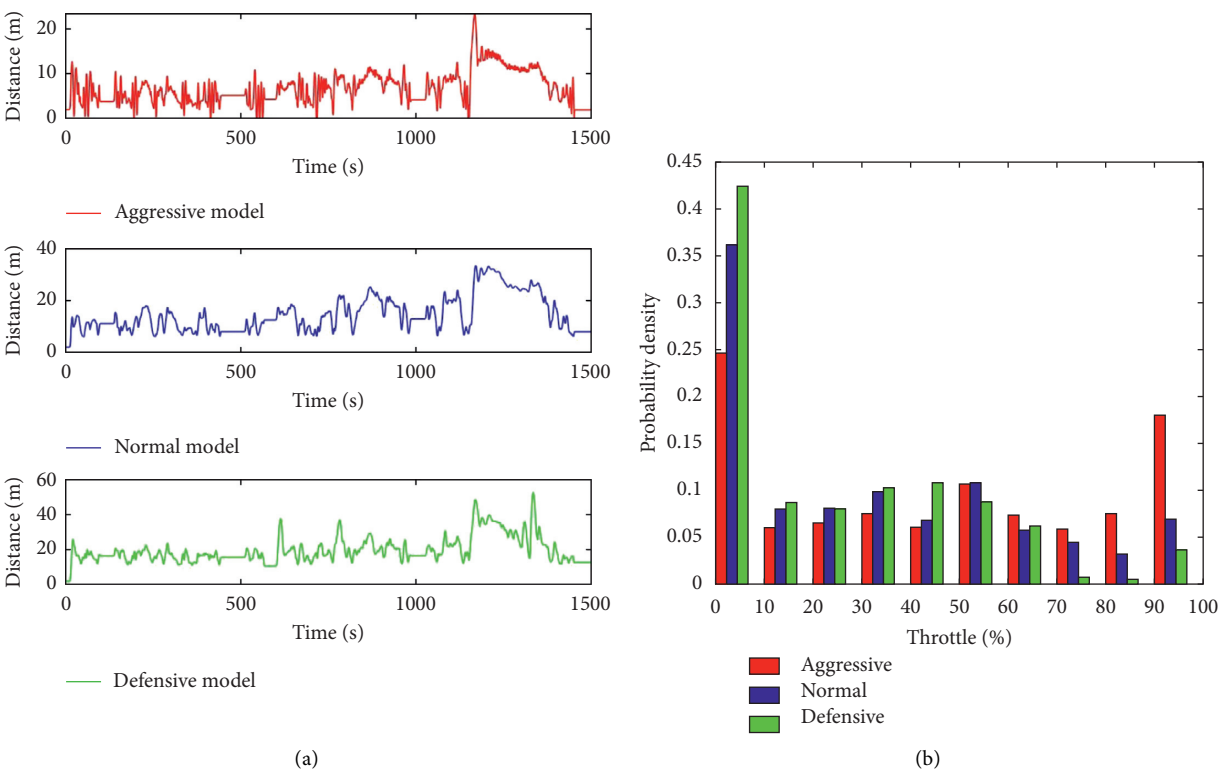


Figure 10: Driving style comparison. (a) Headway distance. (b) Throttle pedal demand.

Meanwhile, the probability density distributions of throttle pedal demands were also selected to investigate the difference between the three driving styles, as it can directly reveal the driver's intention. From Figure 10(b), it can be noted that the defensive model possesses a larger proportion of smaller throttle demands, while the aggressive model shows a greater tendency of harsh acceleration. Meanwhile, the calibrated normal model's probability density was almost equally distributed, with a smaller proportion between 70% and 90%. Although the changing tendencies of probability density among the three driving styles were not consistent within some throttle segments, it can be noted that the aggressive driving shows a preference for larger throttle values and hence harsher accelerations. Moreover, as jerks were also widely used for driving style classification [49], the average jerks of these three models were hence computed using the corresponding speed profiles to further validate these three relative driving style models, and they were normalized using the jerk of the WLTC drive cycle. The normalized jerk values for aggressive, normal, and defensive models were 2.1343, 1.2237, and 0.9046, respectively. As a larger jerk denotes harsher accelerations and decelerations, the obtained results also demonstrated that three relative driving styles' modelling was successful.

The accumulated fuel consumption of the three models was also estimated during the simulation. It was found that the difference between the established driving style models in the Low speed segment is not prominent. This could be caused by the relatively small following distance, as aggressive driving is restricted in this scenario. Meanwhile, the variations of fuel consumption become more significant in the Medium and High speed segments, revealing the potential influence of driving style on fuel consumption. The estimated total fuel consumption of the aggressive driver model was about 1.16 litre, which was approximately 6% more when compared with the defensive driver model.

4.4. Limitation and Future Work. While the proposed method has achieved promising performance in simulating humanized driving style, some limitations are identified. It should be noted that although the proposed method shows higher robustness to incomplete data during the experiment, its performance can be affected by the suitability of expert knowledge. Therefore, the proposed modelling method can perform better when accurate expert knowledge can be obtained from the corresponding human driver.

As the future work, more environmental inputs, such as road gradient, time, and weather, can be incorporated to maximize the similarity to human drivers. Moreover, different test drivers' naturalistic driving data will be collected and used to formulate more accurate driving style variance. Furthermore, other calibration approaches will also be employed to improve the fuzzy logic controller's calibration process. Humanized driver models trained using the proposed approach can also be integrated with the decision-making process when designing advanced driver assistance system (ADAS) and the control strategy of autonomous vehicles (AVs) [50].

5. Conclusion

The primary aim of evaluating different driving styles by developing respective driver models was achieved. A data-driven approach to develop the humanized driver model was developed. A VW Sharan was instrumented to facilitate the data collection. The vehicle state information was logged using a Rebel data logger, and headway distance was obtained from a continental radar and a Nextbase dashcam. Aside from the naturalistic driving data collection, the simulation was entirely based on Simulink. A vehicle model denoting VW Sharan was first developed and validated using WLTC drive cycle data collected from chassis dynamometer experiments. Afterwards, a fuzzy logic driver model was established, and the membership functions were calibrated using naturalistic driving data. It was found that the established driver model was capable of simulating three relative driving styles. A PID-based driver model was also developed for comparison. Those driver models created with the proposed approach were found to perform better in preserving the driving style of corresponding human drivers. The influence of driving styles on fuel consumption was also compared with the obtained models. It was found that the aggressive model can consume approximately 6% more fuel than the defensive model.

This study's major contribution was to theoretically propose a novel approach for humanized driver modelling, which is capable of simulating different driving styles using environmental inputs from human driver perception. Moreover, an anchored procedure to standard drive cycle test was proposed, which can incorporate the driving style difference into these tests. Furthermore, the difference in driving style and their influence on fuel consumption was evaluated in a unified comparative environment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Elander, R. West, and D. French, "Behavioral correlates of individual differences in road-traffic crash risk: an examination of methods and findings," *Psychological Bulletin*, vol. 113, no. 2, pp. 279–294, 1993.

[2] F. Saad, "Behavioural adaptations to new driver support systems: some critical issues," in *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 288–293, The Hague, Netherlands, October 2004.

[3] H. A. Deery, "Hazard and risk perception among young novice drivers," *Journal of Safety Research*, vol. 30, no. 4, pp. 225–236, 1999.

[4] T. Lajunen and T. Özkan, "Self-report instruments and methods," in *Handbook of Traffic Psychology*, B. Porter, Ed., Elsevier, Amsterdam, Netherlands, 2011.

[5] L. M. B. Kleisen, *The relationship between thinking and driving styles and their contribution to young driver road safety*, Dissertation (PhD), University of Canberra, Canberra, Australia, 2011.

[6] C. Macadam, Z. Bareket, P. Fancher, and R. Ervin, "Using neural networks to identify driving style and headway control behavior of drivers," *Vehicle System Dynamics*, vol. 29, no. 1, pp. 143–160, 1998.

[7] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 1609–1615, Washington DC, USA, October 2011.

[8] O. Taubman-Ben-Ari, M. Mikulincer, and O. Gillath, "The multidimensional driving style inventory-scale construct and validation," *Accident Analysis & Prevention*, vol. 36, no. 3, pp. 323–332, 2004.

[9] G. Meiring and H. Myburgh, "A review of intelligent driving style analysis systems and related artificial intelligence algorithms," *Sensors*, vol. 15, no. 12, pp. 30653–30682, 2015.

[10] H. Y. Zhao, H. Zhou, C. F. Chen, and J. M. Chen, "Join driving: a smart phone-based driving behaviour evaluation system," in *Proceedings of the 2013 IEEE Global Communications Conference*, pp. 48–53, Atlanta, GA, USA, December 2013.

[11] F. Sagberg, S. Selpi, G. F. Bianchi Piccinini, and J. Engström, "A review of research on driving styles and road safety," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 57, no. 7, pp. 1248–1275, 2015.

[12] A. Aljaafreh, N. Alshabatat, and M. S. N. Al-Din, "Driving style recognition using fuzzy logic," in *Proceedings of the 2012 IEEE International Conference on Vehicular Electronics and Safety*, pp. 460–463, Istanbul, Turkey, July 2012.

[13] M. S. N. Al-Din, A. Aljaafreh, N. Albdour, and M. Saleh, "Driving styles recognition using decomposed fuzzy logic system," *International Journal of Electrical, Electronics & Computer Systems*, vol. 16, no. 1, pp. 1–5, 2013.

[14] D. Dörr, D. Grabengiesser, and F. Gauterin, "Online driving style recognition using fuzzy logic," in *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems*, pp. 1021–1026, Qingdao, China, October 2014.

[15] J. E. Meseguer, C. T. Calafate, J. C. Cano, and P. Manzoni, "Characterizing the driving style behaviour using artificial intelligence techniques," in *Proceedings of the 38th IEEE Conference On Local Computer Networks*, pp. 1–3, Sydney, Australia, October 2013.

[16] W. S. Wang and J. Q. Xi, "A rapid pattern-recognition method for driving styles using clustering-based support vector machines," in *Proceedings of the 2016 American Control Conference*, pp. 5270–5275, Boston, MA, USA, July 2016.

[17] Y. Feng, S. Pickering, S. Pickering, E. Chappell, P. Iravani, and C. Brace, "A support vector clustering based approach for driving style classification," *International Journal of Machine Learning and Computing*, vol. 9, no. 3, pp. 344–350, 2019.

[18] G. F. Li, Y. Y. Chen, D. P. Cao et al., "Extraction of descriptive driving patterns from driving data using unsupervised algorithms," *Mechanical Systems and Signal Processing*, vol. 156, Article ID 107589, 2021.

[19] G. Li, S. E. Li, B. Cheng, and P. Green, "Estimation of driving style in naturalistic highway traffic using maneuver transition probabilities," *Transportation Research Part C: Emerging Technologies*, vol. 74, pp. 113–125, 2017.

[20] J. L. Deffenbacher, E. R. Oetting, and R. S. Lynch, "Development of a driving anger scale," *Psychological Reports*, vol. 74, no. 1, pp. 83–91, 1994.

[21] J. C. F. de Winter and D. Dodou, "The driver behaviour questionnaire as a predictor of accidents: a meta-analysis," *Journal of Safety Research*, vol. 41, no. 6, pp. 463–470, 2010.

[22] S. Amado, E. Ankan, G. Kaca, M. Koyuncu, and B. N. Turkan, "How accurately do drivers evaluate their own driving behavior? an on-road observational study," *Accident Analysis & Prevention*, vol. 63, pp. 65–73, 2013.

[23] J. Tulusan, T. Staake, and E. Fleisch, "Providing eco-driving feedback to corporate car drivers: what impact does a smartphone application have on their fuel efficiency?" in *Proceedings of the 14th ACM International Conference on Ubiquitous Computing*, pp. 212–215, Pittsburgh, PA, USA, September 2012.

[24] M. Staubach, N. Schebitz, F. Köster, and D. Kuck, "Evaluation of an eco-driving support system," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 27, pp. 11–21, 2014.

[25] M. J. M. Sullman, L. Dorn, and P. Niemi, "Eco-driving training of professional bus drivers - does it work?" *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 749–759, 2015.

[26] H. Liu, F. Sun, D. Guo, B. Fang, and Z. Peng, "Structured output-associated dictionary learning for haptic understanding," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1564–1574, 2017.

[27] W. A. Tillmann and G. E. Hobbs, "The accident-prone automobile driver," *American Journal of Psychiatry*, vol. 106, no. 5, pp. 321–331, 1949.

[28] H. B. Gao, G. Y. Shi, G. T. Xie, and B. Cheng, "Car-following method based on inverse reinforcement learning for autonomous vehicle decision-marking," *International Journal of Advanced Robotic Systems*, vol. 16, no. 6, pp. 1–11, 2018.

[29] T.-Q. Tang, J. He, S.-C. Yang, and H.-Y. Shang, "A car-following model accounting for the driver's attribution," *Physica A: Statistical Mechanics and Its Applications*, vol. 413, pp. 583–591, 2014.

[30] Y. Zhang, P. Ni, M. Li, H. Liu, and B. Yin, "A new car-following model considering driving characteristics and preceding vehicle's acceleration," *Journal of Advanced Transportation*, vol. 2017, pp. 1–14, 2017.

[31] L.-H. Luo, H. Liu, P. Li, and H. Wang, "Model predictive control for adaptive cruise control with multi-objectives: comfort, fuel-economy, safety and car-following," *Journal of Zhejiang University Science A*, vol. 11, no. 3, pp. 191–201, 2010.

[32] C. Zhang and A. Vahidi, "Predictive cruise control with probabilistic constraints for eco driving," in *Proceedings of the ASME 2011 Dynamic Systems and Control Conference*, pp. 1–6, Arlington, TX, USA, November 2011.

[33] C. Su, W. W. Deng, R. He, J. Wu, and Y. D. Jiang, "Personalised adaptive cruise control considering drivers' characteristics," 2018.

[34] H. Hao, W. Ma, and H. Xu, "A fuzzy logic-based multi-agent car-following model," *Transportation Research Part C: Emerging Technologies*, vol. 69, pp. 477–496, 2016.

[35] G. N. Bifulco, F. Simonelli, and R. D. Pace, "Experiments toward an human-like adaptive cruise control," in *Proceedings of the 2008 IEEE Intelligent Vehicles Symposium*, pp. 919–924, Eindhoven, Netherlands, June 2008.

[36] B. Shi, W. Meng, H. Liu, J. Hu, and L. Xu, "A normalized approach for evaluating driving styles based on personalized driver modeling," *Lecture Notes in Electrical Engineering*, in

Proceedings of the 2014 SAE-China Congress & China Automotive Engineering and Manufacturing Expo, pp. 433–444, Shanghai, China, October 2014.

[37] B. Shi, L. Xu, J. Hu et al., "Evaluating driving styles by normalizing driving behavior based on personalized driver modeling," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 12, pp. 1502–1508, 2015.

[38] B. Shi, L. Xu, H. Jiang, and W. Meng, "Comparing fuel consumption based on normalised driving behaviour: a case study on major cities in China," *IET Intelligent Transport Systems*, vol. 11, no. 4, pp. 189–195, 2017.

[39] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F. Y. Wang, "Capturing car-following behaviors by deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 910–920, 2017.

[40] H. Gao, G. Shi, G. Xie, and B. Cheng, "Car-following method based on inverse reinforcement learning for autonomous vehicle decision-making," *International Journal of Advanced Robotic Systems*, vol. 15, no. 6, pp. 1–11, 2018.

[41] Y. D. Jiang, W. W. Deng, J. S. Wang, and B. Zhu, *Studies on Drivers' Driving Styles Based on Inverse Reinforcement Learning*, SAE Internationals, Warrendale PA, USA, 2018.

[42] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 348–368, 2018.

[43] Y. X. Feng, S. Pickering, E. Chappell, P. Iravani, and C. Brace, "Distance estimation by fusing radar and monocular camera with kalman filter," 2017.

[44] P. Volkswagen Sharan (2010 onwards) Specs & Dimensions [Online]. Peterborough: Parkers. Available from: http://www.parkers.co.uk/volkswagen/sharan/estate-2010/20-tdi-cr-bluemotion-tech-(140bhp)-s-5d/specs.

[45] H. B. Yadav and D. K. Yadav, "A method for generating membership function from numerical data," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 5, pp. 2227–2233, 2015.

[46] M. Tutuianu, A. Marotta, H. Steven et al., *Development of a World-Wide Worldwide Harmonized Light Duty Driving Test Cycle (WLTC)*, UNECE, Geneva, Swizerland, 2014.

[47] D. J. Cole, "A path-following driver-vehicle model with neuromuscular dynamics, including measured and simulated responses to a step in steering angle overlay," *Vehicle System Dynamics*, vol. 50, no. 4, pp. 573–596, 2012.

[48] E. Tzirakis and F. Zannikos, "Impact of driving styles on fuel consumption and exhaust emissions: defensive and aggressive driving style," in *Proceedings of the 10th International Conference on Environmental Science and Technology*, pp. 1497–1504, Cos island, Greece, September 2007.

[49] Y. L. Murphey, R. Milton, and L. Kiliaris, "Driver's style classification using jerk analysis," in *Proceedings of the 2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, pp. 23–28, Nashville, TN, USA, April 2009.

[50] G. F. Li, Y. F. Yang, T. R. Zhang et al., "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 122, Article ID 102820, 2021.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9963394, 10 pages
https://doi.org/10.1155/2021/9963394

Research Article

Prediction and Analysis of Train Passenger Load Factor of High-Speed Railway Based on LightGBM Algorithm

Bing Wang,[1] **Peixiu Wu**,[1] **Quanchao Chen**,[1,2,3] **and Shaoquan Ni**[1,2,3]

[1]*School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, China*
[2]*National and Local Joint Engineering Laboratory of Comprehensive Intelligent Transportation, Southwest JiaoTong University, Chengdu, China*
[3]*National Engineering Laboratory of Integrated Transportation Big Data Application Technology, Chengdu, China*

Correspondence should be addressed to Shaoquan Ni; shaoquanni@163.com

Received 3 April 2021; Revised 9 May 2021; Accepted 2 June 2021; Published 15 June 2021

Academic Editor: Chi-Hua Chen

In order to improve the prediction accuracy of train passenger load factor of high-speed railway and meet the demand of different levels of passenger load factor prediction and analysis, the influence factor of the train passenger load factor is analyzed in depth. Taking into account the weather factor, train attribute, and passenger flow time sequence, this paper proposed a forecasting method of train passenger load factor of high-speed railway based on LightGBM algorithm of machine learning. Considering the difference of the influence factor of the passenger load factor of a single train and group trains, a single train passenger load factor prediction model based on the weather factor and passenger flow time sequence and a group of trains' passenger load factor prediction model based on the weather factor, the train attribute, and passenger flow time sequence factor were constructed, respectively. Taking the train passenger load factor data of high-speed railway in a certain area as an example, the feasibility and effectiveness of the proposed method were verified and compared. It is verified that LightGBM algorithm of machine learning proposed in this paper has higher prediction accuracy than the traditional models, and its scientific and accurate prediction can provide an important reference for the calculation of passenger ticket revenue, operation benefit analysis, etc.

1. Introduction

High-speed railway has become the main transportation mode for passengers' mode of transportation for passengers. According to the relevant statistical data, in 2019, the passenger volume of the national railway reached 3.57 billion, of which the passenger volume of multiple unit train was 2.29 billion, accounting for 64.15%. High-speed railway is a significant driver of railway passenger operation revenue and passenger flow growth, and its profit and loss analysis is critical to train operation and operation decision, and the passenger load factor is used as a direct measure of train operation efficiency and a momentous basis for calculating passenger ticket revenue. Scientific and accurate prediction of the train passenger load factor can provide significant reference for train operation scheme, ticket revenue calculation, operational benefit analysis, and so on and so forth [1].

Passenger load factor prediction of passenger trains is usually based on historical data of passenger tickets; the traditional method is to input the train information into the electronic form, use manual to process, classify, and estimate the passenger load factor, and to form a decision table [2]. Nevertheless, there are some problems such as substantial error and inconsistent decision information. At present, there are quite a few research studies on the passenger load factor prediction of high-speed railway multiple units. Different scholars use a variety of model methods to predict, such as multiple regression model, time series model, neural network model, decision tree model, gray theory model, and integrated learning algorithm model. Aiming at the competitive relationship between high-speed train and air transport, Wang et al. [3] proposed a prediction model of the passenger load factor of the high-speed railway trains based on Adaboost-CART algorithm from the perspective of the

impact of air fare level and dynamic fluctuation on high-speed railway trains passenger flow. From the perspective of train attributes, Zhang et al. [4] proposed a classification and prediction model of the passenger train load factor based on random forest algorithm. On the basis of analyzing the influencing factors of the train passenger load factor, Xu and Nie [5] established a BP neural network prediction model of train passenger load factor, considering the factors of train attributes and the operation period. Based on two single prediction models, ARIMA model and BP neural network, Zhang and Bai [6] constructed a linear combination prediction model of railway passenger load factor according to the principle of minimum sum of square errors [7]. In the research of machine learning prediction, quite a few studies use machine learning algorithms for short-term traffic flow prediction [1, 8, 9] and use LightGBM and XGBoost for prediction and classification [10–15]. Dong et al. [16] established a short-term traffic flow prediction model based on XGBoost algorithm. After analyzing the vehicle speed, road, and weather features in the course of the operation of the bus, Wang et al. [17] established a prediction model of bus travel time based on LightGBM algorithm. Huang et al. [18] constructed deep belief neural network based on multitask learning to predict traffic volume [19]. Zhang et al. [2] constructed a short-term traffic flow prediction model based on the fusion algorithm of XGBoost and LightGBM.

In the study of the passenger load factor or passenger volume forecast, various models are mainly used to predict the historical passenger load factor or passenger volume or passenger volume, and the rules of generating target variables are rarely obtained according to the attributes of trains [20–23]. In this paper, for high-speed railway trains, consider the influence factors such as train attributes, historical weather, and passenger flow sequence, and a single train passenger load factor prediction model and a group train passenger load factor prediction model based on the LightGBM algorithm are proposed, which can provide decision-making basis for ticket revenue calculation and operation benefit analysis.

1.1. Influence Factors. The passenger load factor of high-speed railway is an index reflecting the utilization degree of passenger carrying capacity. It is the ratio of passenger turnover to the total number of passenger kilometers, which is expressed as a percentage of the average number of passenger per kilometer [24]. The passenger load factor data comes from the analysis and statistics of passenger flow; in essence, the passenger flow determines the passenger load factor, and the influencing factors of passenger travel choice are the factors that affect the train occupancy rate. Hence, the analysis of influencing factors of the passenger load factor is the analysis of passenger flow travel choice.

From the perspective of demand and the macro-distribution of passenger flow, the spatial distribution of passenger flow is determined by the regional economic development level, population, and function orientation along the high-speed railway. In a period of time, the macrofactors such as regional economic development level are relatively stable, so the spatial distribution of passenger flow is also at a relatively stable level. Passenger travel has evident time preference for specific travel behavior, and the departure and arrival time of a train will directly affect the train's load factor. Simultaneously, travel time and weather will affect the choice of the travel mode. For different time's nodes, on weekdays, most passengers travel mainly for business; nevertheless, on weekends and holidays, most passengers travel for tourism, family visits, etc. Therefore, the difference in travel time will also affect the load factor of the train.

From the perspective of transportation supply, firstly, the distribution of passenger flow direction is unbalanced, so the train operation direction will affect the train passenger load factor; the number of trains running between OD, namely, the service frequency of trains between OD is one of the main factors affecting the choice of passengers; furthermore, the departure and arrival time, running mileage, station of the way, train capacity, and type of the train will all affect the choice of passenger travel, thus affecting the load factor of the train [21].

By and large, the influencing factors of the train passenger load factor can be divided into internal factors and external factors, as shown in Figure 1. In a period of time, the regional economic level, population, and function orientation of city is relatively stable; furthermore, the main factors affecting the passenger load factor of high-speed railway are the direction of train operation, service frequency of OD, train attributes, weather, and travel time [14]. To this, this paper mainly considers attributes of the train, weather, and travel time.

2. Prediction Model

The passenger load factor prediction model of high-speed railway is mainly composed of data acquisition, data processing, feature engineering, model training, and prediction.

(1) Data acquisition: the historical weather data, train attribute data, and passenger load factor data are obtained from various ways, and the original data is formed by data fusion.

(2) Data processing: the types and dimensions of data variables are inconsistent; therefore, it is necessary to transform the categories and features of the original data before data modeling so that the data can meet the requirements of algorithm data structure. And, data processing mainly includes data transformation and data cleaning, such as removing the character "°C" from the maximum and minimum temperature of the weather data. For the train with few times that are temporarily operating or have been suspended, it should be deleted (high-speed railway below 60 times in this paper have been deleted); as there is a great difference in the change of the passenger load factor in each operation period, this thesis mainly selects weekday data for research.

(3) Feature engineering: average temperature feature, decomposition date feature, and multiclass

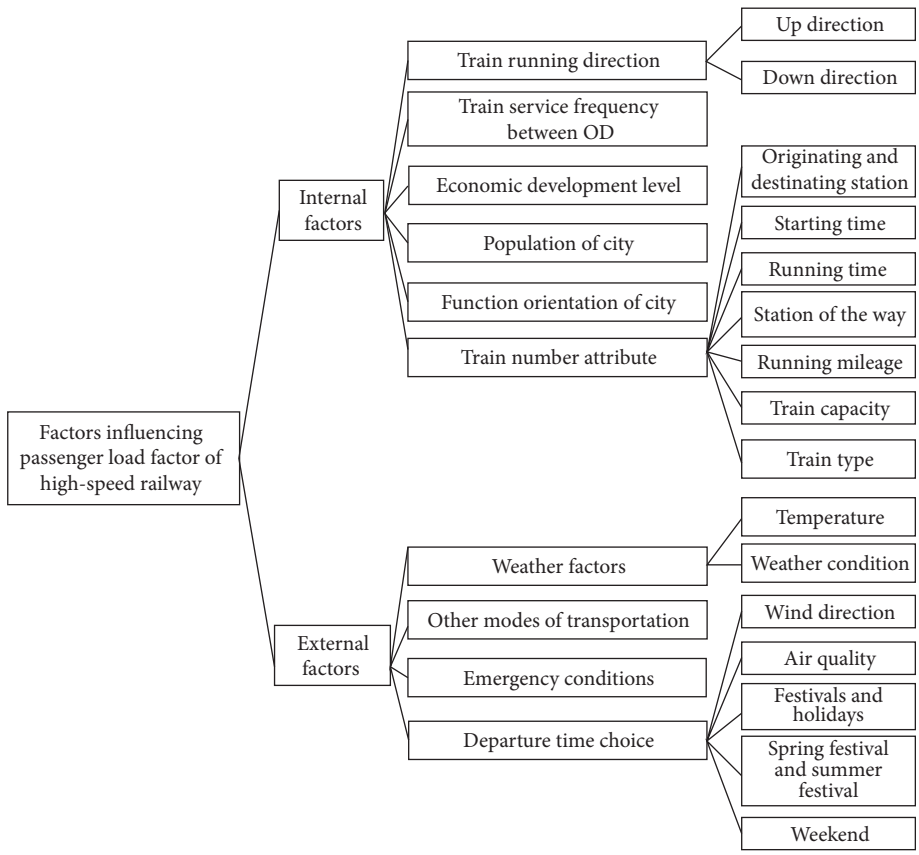


Figure 1: Influencing factors of train passenger load factor.

classification variable code are constructed. For train attribute, weather, and time characteristic data, the following characteristic engineering processing is carried out:

① The minimum and maximum temperatures are replaced by the characteristic average temperature (Avgtemperature)

② Constructing the features of passenger flow time sequence, new features DayOfWeek, WeekOfYear, Month, and Day are constructed by decomposing date features, and it denotes the days of a week, the week of a year, the months of a year, and the days of a month, respectively

③ LabelEncoder features mileage, capacity, weather, wind turbines (WIntensity), departure time (DepaTime), run time (OperTime), and turn discrete variables into multiclass continuous numerical variables

All feature engineering obtained after data preprocessing, including historical weather and train attributes of the passenger rate data, are shown in Table 1.

(4) Model training and prediction: after feature engineering, the sample data set is constructed and divided into the training set and the test set. In the meantime, the model is trained and tested. The framework of the prediction model is shown in Figure 2.

For a single train, the attributes of the train are fixed, and the main factors affecting its passenger load factor are weather and time series characteristics. For group trains, the attributes of the train are one of the cardinal factors affecting the passenger load factor, therefore, the prediction of passenger load factor prediction of group trains, train attributes, weather features, and time-series characteristics need to be considered.

3. Case Analysis

To verify the effectiveness of the single train passenger load factor prediction model and group train passenger load factor prediction model based on LightGBM algorithm proposed in this paper, the passenger load factor data of all down directions from station A of a high-speed railway in the area as an example is taken. And, the load factor data of high-speed railway comes from all trains that departed from station A in the downward direction from October 9, 2017, to September 30, 2019, (provided by Station A); historical weather data is obtained from the 2345 weather forecast network through *Python* (http://tianqi.2345.com/). The target data covers 722 days of historical data of station A from October 9, 2017, to September 30, 2019; train attribute data includes the train number/type, departure station, terminal station, departure time, arrival time, traveling time, stop station, and ticket price (https://www.12306.cn). Based on machine LightGBM algorithm, this paper forecasts the

Table 1: Features of train passenger load factor data.

Feature	Variable type
Mileage	Multiple categorical variable
Capacity	Multiple categorical variable
Avgtemperature	Continuous numerical value
Weather	Multiple categorical variable
WIntensity	Multiple categorical variable
DepaTime	Multiple categorical variable
AQI	Multiple categorical variable
OperTime	Multiple categorical variable
DayOfWeek	Multiple categorical variable
WeekOfYear	Multiple categorical variable
Month	Multiple categorical variable
Day	Multiple categorical variable
StatNumber (number of intermediate stations)	Continuous numerical value
Load factor	Continuous numerical value

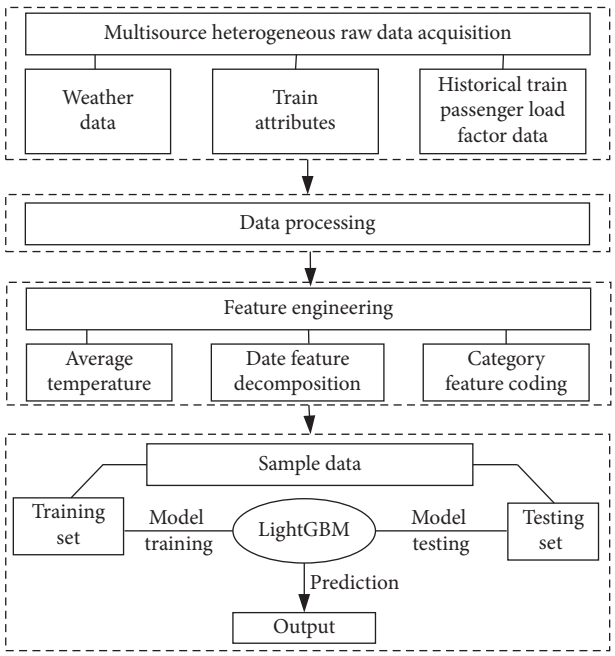


Figure 2: Overall framework of prediction model.

data of train passenger load factor, and the train load factor data of the first 600 days is taken as the training set and the last 67 days is taken as the test set.

3.1. Passenger Load Factor Prediction of a Single Train Based on LightGBM Algorithm

3.1.1. Model Train. The passenger load factor of a train is predicted based on the LightGBM algorithm and, in comparison, with XGBoost and ARIMA algorithm. During the training in the training set, use LightGBM. Cv () function to optimization of 10-fold cross validation parameters; set the learning_rate to 0.01 and adjust the parameters {n_estimators, num_leaves, bagging_ fraction, bagging_freq, feature_fraction, max_bin, min_data_in_leaf, lambda_11, lambda_12, min_split_ gain, max_depth} in turn; finally, a set of optimal parameters is obtained, and then, fine adjustment is made. MAE is used as the index of performance evaluation in the training process:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|, \tag{1}$$

where the actual value of train passenger load factor is y_i, and $\hat{y}_i$ is the forecast value.

3.1.2. Result Analysis. After the optimal parameters are trained by the training set model, the visual fitting process is shown in Figure 3 and compared with XGBoost algorithm.

It can be seen from Figure 3 that the model cannot be well identified and fitted at the mutation point of the passenger load factor, but it can fit other relatively stable points well. Meanwhile, considering that the passenger load factor sequence is a kind of time sequence, in order to further verify the effectiveness of the model constructed in this paper, the ARIMA model is selected for the comparison test.

For the ARIMA (p, d, q) model, through ADF unit root test, Ljung-Box test, ACF chart of autocorrelation coefficient, and PACF chart of partial autocorrelation coefficient combined with AIC and BIC minimum as the target order, ARIMA (7, 8) is determined as the final model, and the visualization of the fitting of the model is shown in Figure 4, where the red sequence is the true values and the yellow sequence is the fitting values.

The trained model is used to predict the test set of LightGBM and XGBoost and rolling prediction ARIMA, and then, the MAE of three models in the training set and test set is obtained, as shown in Table 2. Therefore, LightGBM has the best prediction performance; the ARIMA model has the worst fitting, the lowest prediction accuracy, and the rolling prediction needs to increase the actual value in each step and then retrain, which is not suitable for multistep prediction. The LightGBM and XGBoost prediction results are compared with the true values, as shown in Figure 5.

The LightGBM model is used to predict the passenger load factor of the selected train and visualize the importance of its features, as shown in Figure 6.

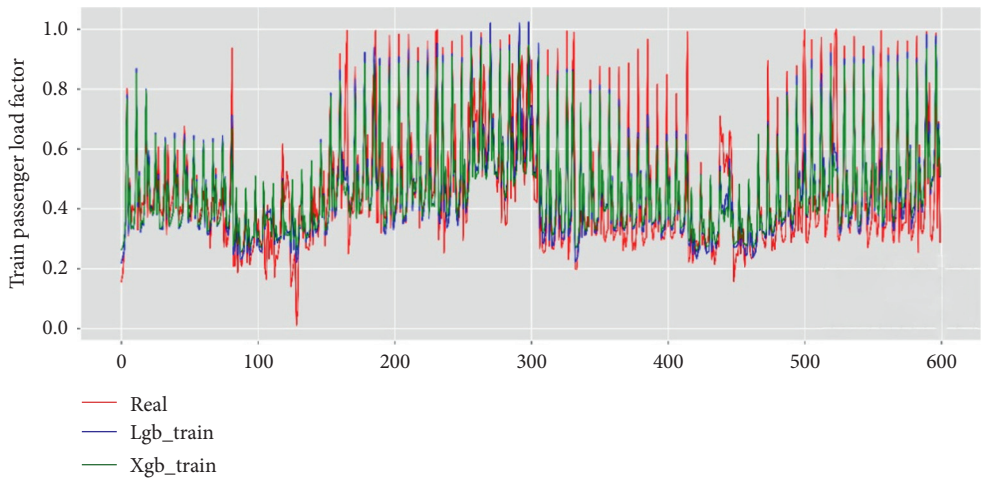


Figure 3: Comparison between LightGBM and XGBoost.

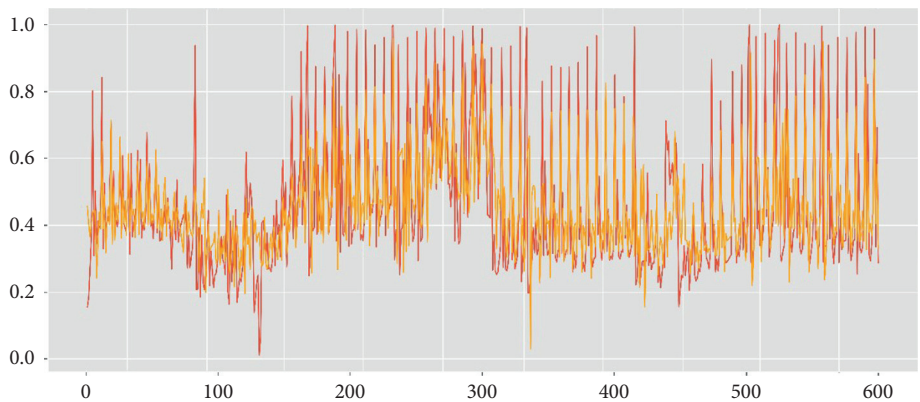


Figure 4: Comparison of ARIMA model predicted value and real value.

Table 2: Performance comparison of LightGBM, XGBoost, and ARIMA models.

Algorithm	Train set MAE	Test set MAE
LightGBM	0.071622	0.082692
XGBoost	0.080532	0.093274
ARIMA (7, 8)	0.113674	0.135241

It can be seen from Figure 6 that the characteristics of the passenger rate of the train are sorted by importance are WeekOfYear, DayOfWeek, Avgtemperature, Day, Weather, Month, and AQIlevel (air quality level), of which the most important is WeekOfYear and the least important feature is AQIlevel.

3.2. Group Train Passenger Load Factor Prediction Based on LightGBM Algorithm. Before the prediction of the passenger load factor of group train, the histogram of train passenger load factor to be predicted is drawn to check the distribution of its value. The x-axis represents the 0 to 1 passenger load factor divided into 100 cells and the y-axis represents the statistics for the cells, as shown in Figure 7; according to the histogram, the passenger load factor of the sample is a long tail distribution, and it has a large imbalance.

LightGBM algorithm can set the parameters of data acquisition in the course of training, compared with other traditional machine learning algorithms; it ensures that the data acquisition of training keeps the original proportion, which is more suitable for dealing with the issue of unbalanced sample distribution. LightGBM algorithm has designed the parameter class_Weight, passing the value "balanced" to this parameter, and it will automatically calculate various weights according to the classification label value. The problem of unbalanced sample distribution can be adjusted, which is helpful to the convergence of training when the samples are unbalanced. Therefore, to ensure that the passenger load factor of group train is predicted under small error, the prediction of passenger load factor of the group train with 10 classifications is constructed in [7], as shown in Figure 8, and good classification results are obtained.

3.2.1. Model Train. The data processed by feature engineering are divided into the training set and test set, and the data of test set is the last month. In the given parameter space, the cross-validation function "LightGBM. cv ()"of LightGBM official website is used to optimize the parameters. The parameters that need to be optimized for

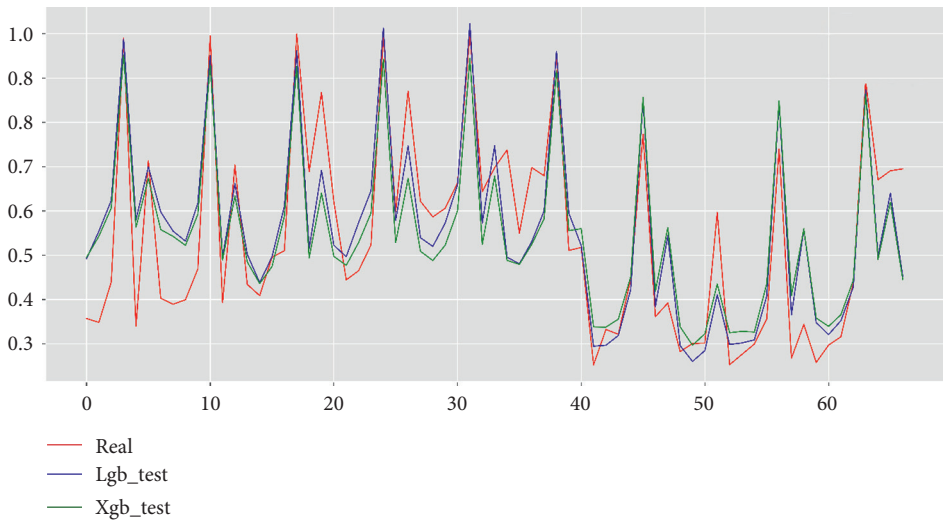


Figure 5: Prediction results of LightGBM and XGBoost.

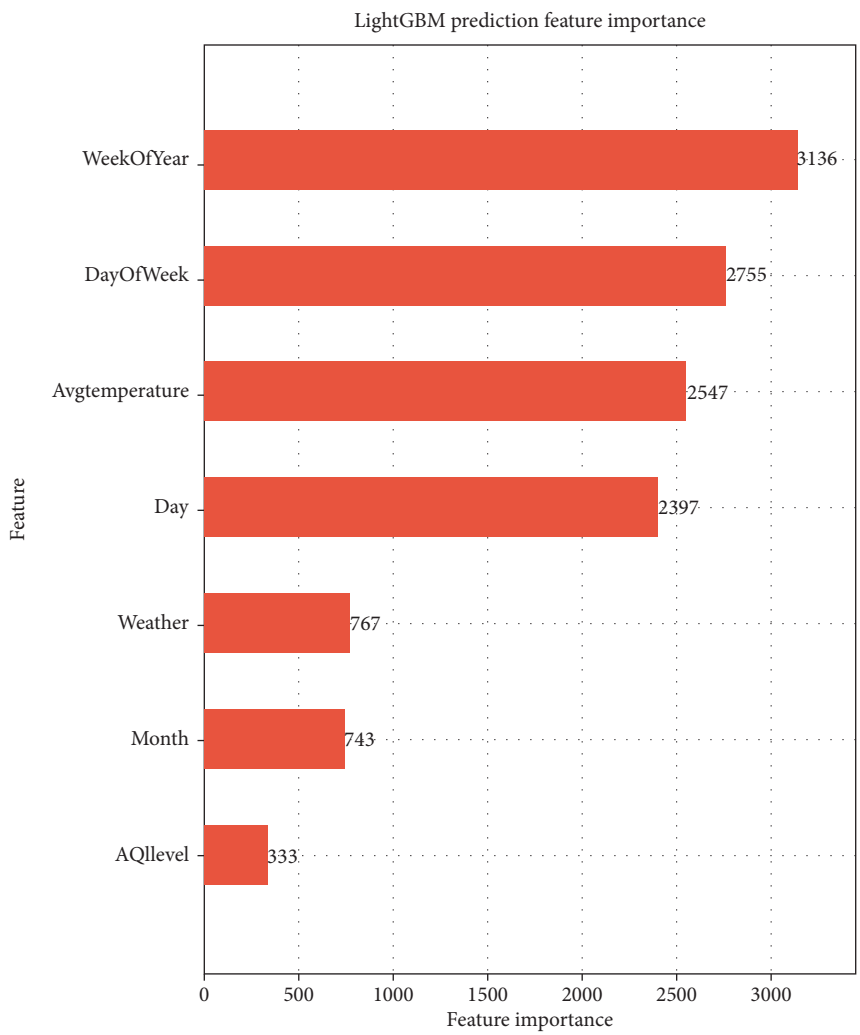


Figure 6: Importance of prediction model features.

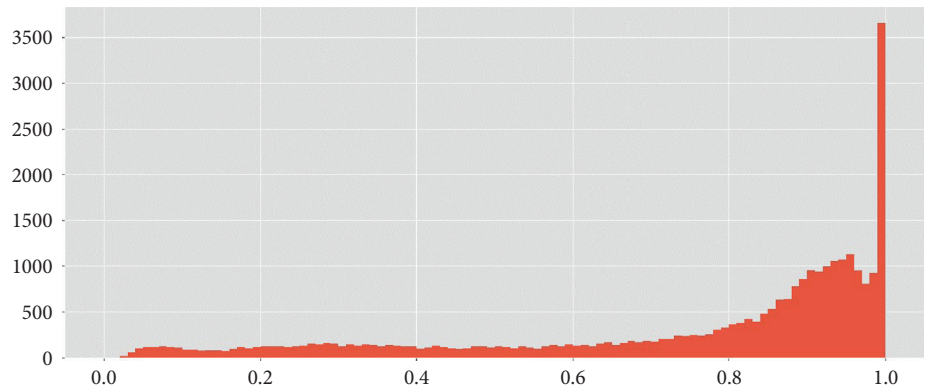


Figure 7: Distribution of load factor data.

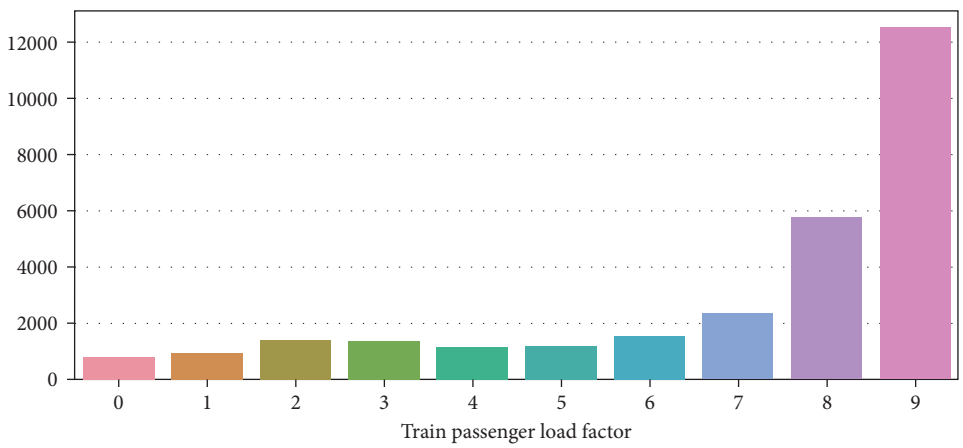


Figure 8: 10 classifications of load factor data.

LightGBM function are {"num_leaves","max_bin", "min_data_in_leaf", "feature_fraction", "bagging_fraction", "bagging_freq","lambda_11", "lambda_11", "min_-split_gain", "learing_rate"}. After the optimal combination of the parameters is obtained, the optimal model is retrained in the training set, and finally, the model is used to predict the test set.

For binary classification, 1 is used as a positive example and 0 as a negative example, and the four classifications are defined as follows:

(1) TP (true positive) indicates the number of samples that the actual values and the predictions are positive examples

(2) FP (false positive) indicates the number of samples that the actual values are negative example but predicted to be positive

(3) FN (false negative) indicates the number of samples that the actual values are positive example but predicted to be negative

(4) TN (true negative) indicates the number of samples that the actual values and the predictions are negative examples

For multiclass classification, suppose there are i categories. A similar dichotomous confusion matrix approach can be used to obtain TP, FP, FN, and TN, for each category, which are recorded as TP_i, FP_i, FN_i, and TN_i, respectively. And, for the multiclass classification model, different issues have different evaluation indexes. In the official document of machine learning classification evaluation, Sklearn. metrics. f1_score set the parameter average = "weighted," which can address the problem of unbalanced sample evaluation indexes in multiclassification.

It is denoted as Weighted_F1, and the formula is as follows:

$$\mathrm{Precision}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}, \tag{2}$$

$$\mathrm{Recall}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{3}$$

$$\mathrm{Precision}_W = \frac{\sum_{i=1}^{L} \mathrm{Precision}_i \times \omega_i}{L}, \tag{4}$$

$$\mathrm{Recall}_W = \frac{\sum_{i=1}^{L} \mathrm{Recall}_i \times \omega_i}{L}, \tag{5}$$

$$\mathrm{Weight\_}F1 = \frac{2 \cdot \mathrm{Precision}_w \cdot \mathrm{Recall}_w}{\mathrm{Precision}_w + \mathrm{Recall}_w}, \tag{6}$$

Table 3: The accuracy of LightGBM, XGBoost, and RandomForest forecast classification.

Algorithm	Evaluation index (Weighted_F1)
LightGBM	0.7927
XGBoost	0.7718
RandomForest	0.7160

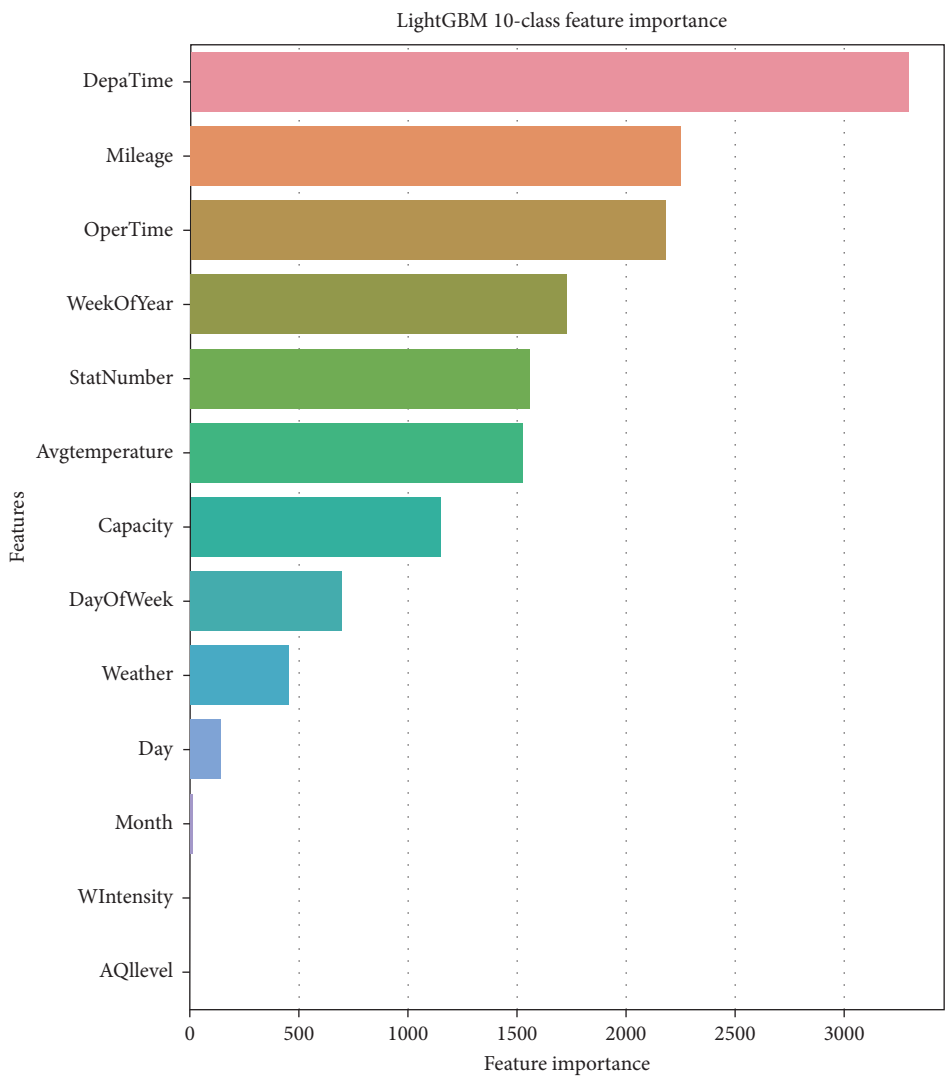


Figure 9: The importance of 10 classification model features.

where TP_i, FP_i, FN_i, and TN_i refer to the calculated value of category i in L multiclass classification and ω_i means the proportion of category i in L multiclassifications. Formula (2) means the ratio of the number of TP_i samples to the number of samples predicted to be positive. That is, the accuracy rate of the calculated value of category i is for the prediction result, which means the probability of actually being category i in all the samples predicted as category i. Formula (3) is expressed as the ratio of the number of samples of TP_i to the number of samples of actual positive examples, which means the recall rate. The precision is for the prediction results, which means the probability of actually being category i among all the samples predicted as category i. Formula (4) is expressed as the ratio of the weight of the accuracy rate to the multicategory. Formula (5) represents the ratio of the weight of the recall rate to the multicategory. Formula (6) is to solve the problem of sample evaluation index imbalance in multicategory.

3.2.2. Result Analysis. The 10 classification model is based on machine learning LightGBM algorithm; the optimal model is optimized after cross validation and the parameter adjustment. For machine learning, the feature_importance function in LightGBM algorithm can calculate, output, and

visualize the importance of each feature. Simultaneously, to verify the prediction model of the group trains' passenger load factor based on the unbalanced 10 classification samples, the XGBoost and RandomForest algorithm are used to compare the prediction results. In XGBoost algorithm, One-Hot Encoder is needed for the features of the categorical variables, and the parameter tuning of RandomForest and XGBoost algorithm is based on *Python* machine learning GridSearchCV function. The prediction results of this paper are shown in Table 3.

It can be seen from Table 3 that the classification consequence of LightGBM algorithm is the best; visualize the importance of the features of LightGBM optimal model, as shown in Figure 9.

It can be seen from Figure 9 that, in the classification and prediction model of the train passenger load factor, the top five important features are DepaTime, Mileage, OperTime, WeekOfYear, and StatNumber, in which DepaTime, Mileage, OperTime, and StatNumber are the train attribute features and WeekOfYear is the time sequence feature of the train passenger load factor. In addition, Capacity is the least important in the features of train attributes, Month is insignificant in the time-series characteristics, and WIntensity and AQIlevel have little influence on a train passenger load factor.

4. Conclusion

In this paper, consider the factors such as train attributes, historical weather, and passenger flow time sequence that affect the passenger load factor of high-speed railway trains; a single train passenger load factor prediction model and a group train passenger load factor prediction model based on LightGBM algorithm are constructed for different prediction requirements and compared with XGBoost, RandomForest and ARIMA algorithm; the feasibility and effectiveness of the prediction model constructed in this paper are verified.

By analyzing the importance of the passenger load factor features' output by the machine learning LightGBM algorithm, the influencing factors of passenger load factor of high-speed railway trains in the region can be obtained. For a train, the crucial factors that affect the passenger load factor are WeekOfYear, DayOfWeek, and average temperature, which are the features of passenger flow time sequence. For high-speed railway trains in a certain area, the main factors affecting the passenger load factor are the attributes of the train, followed by departure time, mileage, and operation time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key R&D Program of China (2017YFB1200702), National Natural Science Foundation of China (project nos. 52072314 and 71971182), Sichuan Science and Technology Program (project nos. 2020YFH0035, 2020YJ0268, 2020YJ0256, 2020JDRC0032, 2021YFQ0001, and 2021YFH0175), Chengdu Science and Technology Plan Research Program (project nos. 2019-YF05-01493-SN, 2020-RK00-00036-ZF, and 2020-RK00-00035-ZF), and Science and Technology Plan of China Railway Corporation (project no. P2018T001 and 2019KY10).

References

[1] J. F. Xu, T. Tang, J. Yan, and Z. Liu, "Prediction of short-term traffic flow based on ensemble learning mechanism," *Journal of Transportation Systems Engineering and Information Technology*, vol. 16, no. 04, pp. 185–190+198, 2016.

[2] M. Zhang, X. Fei, and H. Liu Zhen, "Short-term traffic flow prediction based on combination model of xgboost-lightgbm," in *Proceedings of the 2018 international conference on sensor networks and signal processing (SNSP)*, pp. 322–327, IEEE, Xi'an, China, October 2018.

[3] F. J. Wang, F. J. Wang, Y. C. Wang, and C. Bian, "Bus travel time prediction based on light gradient boosting machine algorithm," *Journal of Transportation Systems Engineering and Information Technology*, vol. 19, no. 02, pp. 116–121, 2019.

[4] Y. Zhang, J. S. Zhu, M. Feng, X. Y. Lu, X. R. Jia, and W. W. Wang, "A study on the model for classifying and predicting occupancy rates of passenger trains," *Rail Way Transport and Economy*, vol. 40, no. 03, pp. 39–45, 2018.

[5] G. Y. Xu and L. Nie, "Research on prediction method for percentage of passenger seats utilization per multiple unit train," *Comprehensive Transportation*, vol. 37, no. 12, pp. 63–67, 2015.

[6] C. L. Zhang and Y. P. Bai, "Application of ARIMA time series and BP neural network combination model in railway passenger rate," *Mathematics in Practice and Theory*, vol. 48, no. 21, pp. 105–113, 2018.

[7] J. S. Jeng-Shyang Pan, W. Qingxiang Feng, Q. X. Lijun Yan, and Y. L. J. Jar-Ferr Yang, "Neighborhood feature line segment for image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 387–398, 2015.

[8] G. Fu, G. Q. Han, F. Lu, and Z. X. Xu, "Short-term traffic flow forecasting model based on support vector machine regression," *Journal of South China University of Technology (Natural Science Edition)*, vol. 41, no. 09, pp. 71–76, 2013.

[9] D. H. Wang, Y. Zhang, and Y. Zhao, "LightGBM: an effective miRNA classification method in breast cancer patients," in *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics (ICCBB 2017)*, pp. 7–11, Association for ComputingMachinery, New York, NY, USA, October 2017.

[10] Q. C. Chen, D. Wen, X. Q. Li et al., "Empirical mode decomposition based long short-term memory neural network forecasting model for the short-term metro passenger flow," *PLoS One*, vol. 14, no. 9, 2019.

[11] M. Gumus and M. S. Kiran, "Crude oil price forecasting using XGBoost," in *Proceedings of the 2017 International Conference*

on Computer Science and Engineering (UBMK), pp. 1100–1103, IEEE, Antalya, Turkey, October 2017.

[12] G. L. Ke, M. Qi, T. Finley et al., "LightGBM: a highly efficient gradient boosting decision tree," in *Proceedings of the Advances in Neural Information Processing Systems 30(NIPS 2017)*, pp. 1–9, San Diego, CA, USA, May 2017.

[13] X. L. Sun, M. X. Liu, and Z. Q. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," *Finance Research Letters*, vol. 32, 2020.

[14] Z. Tian, G. Sun, D. Chen, Z. Qiu, and Y. Ma, "Method for determining the valid travel route of railways based on generalised cost under the syncretic railway network," *Journal of Advanced Transportation*, vol. 2020, Article ID 8287648, 12 pages, 2020.

[15] M. M. Wang, *Research on Short-Term Traffic Flow Forecasting Method Based on Machine Learning*, Chang'an University, Xi'an, China, 2017.

[16] X. C. Dong, T. Lei, S. T. Jin et al., "Short-term traffic flow prediction based on XGBoost," in *Proceedings of 2018 IEEE 7th Data Driven Control and Learning Systems Conference*, pp. 854–859, IEEE, New York, NY, USA, May 2019.

[17] Y. Wang, W. Fang, L. Wang, and B. Xue, "Seating rate prediction for EMU sleeping train based on adaboost-CART model," *China Railway*, vol. 10, pp. 34–38, 2019.

[18] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[19] Y. E. Liu, Y. Z. Zhang, and C. H. Chen, *Review on Deep Learning in Intelligent Transportation Systems*, Springer, Berlin, Germany, 2020.

[20] L. M. Hou and G. F. Ma, "Forecast of railway passenger traffic based on a grey linear regression combined model," *Computer Simulation*, vol. 28, no. 7, pp. 1–3, 2011.

[21] S. Liu, C. S. Yin, D. J. Chen, H. X. Lv, and Q. P. Zhang, "Cascading failure in multiple critical infrastructure interdependent networks of syncretic railway system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 202114 pages, 2021.

[22] Z. Wang, Y. H. Wang, L. M. Jia et al., "The application of improved BP neural network in the prediction of railway passenger volume time serial," *China Railway Science*, vol. 26, no. 2, pp. 127–131, 2015.

[23] G. Y. Xu, *High-speed Train Load Factor Forecast and Breakeven Analysis*, Beijing Jiaotong University, Beijing, China, 2016.

[24] L. B. Deng and F. Shi, "Evaluation index system of passenger train operation plan," *China Railway Science*, vol. 27, no. 3, pp. 106–110, 2006.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5540046, 15 pages
https://doi.org/10.1155/2021/5540046

Research Article

Identifying Incident Causal Factors to Improve Aviation Transportation Safety: Proposing a Deep Learning Approach

Tianxi Dong,[1] Qiwei Yang,[2] Nima Ebadi,[2] Xin Robert Luo,[3] and Paul Rad[4]

[1]*School of Business, Trinity University, One Trinity Place, San Antonio, TX 78212, USA*
[2]*Department of Electrical and Computer Engineering, The University of Texas, San Antonio, TX 78249, USA*
[3]*Anderson School of Management, The University of New Mexico, Albuquerque, NM 87131, USA*
[4]*Department of Information Systems and Cyber Security, The University of Texas, San Antonio, TX 78249,, USA*

Correspondence should be addressed to Tianxi Dong; tdong@trinity.edu

Received 22 January 2021; Revised 24 March 2021; Accepted 25 May 2021; Published 14 June 2021

Academic Editor: Chi-Hua Chen

Aviation is a complicated transportation system, and safety is of paramount importance because aircraft failure often involves casualties. Prevention is clearly the best strategy for aviation transportation safety. Learning from past incident data to prevent potential accidents from happening has proved to be a successful approach. To prevent potential safety hazards and make effective prevention plans, aviation safety experts identify primary and contributing factors from incident reports. However, safety experts' review processes have become prohibitively expensive nowadays. The number of incident reports is increasing rapidly due to the acceleration of advances in information technologies and the growth of the commercial and private aviation transportation industries. Consequently, advanced text mining algorithms should be applied to help aviation safety experts facilitate the process of incident data extraction. This paper focuses on constructing deep-learning-based models to identify causal factors from incident reports. First, we prepare the data sets used for training, validation, and testing with approximately 200,000 qualified incident reports from the Aviation Safety Reporting System (ASRS). Then, we take an open-source natural language model, which is well trained with a large corpus of Wikipedia texts, as the baseline and fine-tune it with the texts in incident reports to make it more suited to our specific research task. Finally, we build and train an attention-based long short-term memory (LSTM) model to identify primary and contributing factors in each incident report. The solution we propose has multilabel capability and is automated and customizable, and it is more accurate and adaptable than traditional machine learning methods in extant research. This novel application of deep learning algorithms to the incident reporting system can efficiently improve aviation safety.

1. Introduction

In the last two decades, we have witnessed rapidly evolving customer expectations and paradigmatic business mergers and acquisitions in the mushrooming development of the aviation industry. In this highly competitive environment, airline companies have increasingly exploited information technologies to turn challenges into business opportunities and support decision-making. Automated decision support technologies remain one of the main challenges in air transportation [1]. Aviation incident reporting and investigation systems are a crucial part of the ongoing digitization of safety efforts. Incidents are anything abnormal that affects or could affect the safety of aviation operations [2]. Unlike accidents, which usually involve fatalities or serious injuries, incidents are much more frequent and less costly than accidents. They are a valuable source of data to help identify potential hazards. Incident reports record various abnormal events and provide reference data to the Federal Aviation Administration, the National Aeronautics and Space Administration, and the National Transportation Safety Board, during the processes of decision-making, procedure design, threat identification, training, and so forth [3]. Since aviation transportation is a highly sophisticated system, many factors, such as human error, aircraft mechanical failure, extreme weather, and unreasonable

company policy, or a combination of them, can result in incidents. Due to the paramount value of incident data, countries and multinational institutes have devoted significant efforts to collecting and storing incident reports for analytical decision-making.

The Aviation Safety Reporting System (ASRS), jointly operated by the FAA and NASA, is one of the leading aviation incident reporting systems and is used extensively in North America. The system receives aviation incident reports submitted by airports, airline companies, pilots, and crews daily. Then the system analyzes and responds to incident reports to identify potential hazards early and prevent aviation accidents. Incident reporting and investigation systems are critical components of safety management in aviation transportation [4]. The information frequently encountered in incident investigations includes the events leading up to the accident, the factors that increased risk, the detection of problems, and the attempts to resolve the problems, all of which can be provided by individuals involved in incidents [5]. The ASRS, a rich and reliable database of information on aviation incidents, is used by NASA and the FAA to evaluate the effectiveness of risk management actions. As a distinctive contribution to safety management, the feedback from incident reporting systems is a vital early-warning tool for decision-makers and planners tasked with improving safety margins in the face of doubled or quadrupled operations [4].

Most of the incident reports are submitted to the ASRS voluntarily. A reporter involved in an incident can fill out an ASRS reporting form anonymously. The narrative is the most informative part of an incident report. The reporter recounts the actual events before, during, and after the incident. Narrative texts mostly describe mechanical failures, observations, behaviors, and weather conditions related to the incident. All submitted ASRS reports are currently manually analyzed and assigned at least one out of sixteen primary factors and no more than four out of sixteen contributing factors by experienced aviation safety analysts [6]. The identification of the primary and contributing factors is a crucial step. The tabular data collected from the reporter includes 96 tabular attributes, such as the reporter's role, qualifications, and experience, type of aircraft involved, type of operator, cabin activity, weather, and many other event-specific details. Unfortunately, based on a random selection of 10,000 incident reports, more than 50% of the incident reports are missing at least half of these attributes, and most of the attributes that are often present, such as date, local time, and state, seem to have little relevance to the causes of the incidents. Thus, the current predicament is that each incident report's narrative text data is the only reliable and informative source to identify the incident-causing factors. Table 1 is an example of a typical ASRS incident report and the conclusions made by human experts (Tables 1 and 2).

The analysis of incident causal factors in the incident reports has been helpful in investigating the root causes of aviation incidents. The research conducted in [7] studied design-induced problems in Flight Management Systems (FMSs) by selecting 99 incident reports related to FMSs from the ASRS. It concluded that a significant number of operational and design-induced problems exist in FMSs, because the user interface of FMSs is not optimally designed. Manufacturers should find a better balance in FMS design between logic and ease of use to reduce the occurrence of errors. Another study [8] used 37 incident reports from the National Transportation Safety Board (NTSB) database to study errors in decision-making in the aviation domain and discussed the nature of such errors, what main factors contribute to them, and what solutions might mitigate them. Reference [9] analyzed the causal factors in aviation maintenance by investigating 3,783 ASRS incident reports related to maintenance incidents. It concluded that individual-related and management-related factors are the most frequent reasons for maintenance error. The nonmaintenance perspective should be given more attention because it can provide abundant information that is usually not included in maintenance personnel reports. To study the multifactor and single-factor effects on human performance in Air Traffic Management (ATM), [10] used over 400 European aviation incident reports related to ATM as their source data. The research concluded that research focusing on single-factor (stress, fatigue, communication, etc.) effects on human performance is poorly suited to the complexities of contemporary ATM, because incident reports often indicated multifactor cooccurrences. In sum, a collection of aviation safety research and analysis has relied on incident reports and their conclusions about causal factors. At present, the ASRS heavily depends on human experts to identify the causal factors. However, the increasing number of incident reports submitted every day, due to the rapid growth of the aviation industry, has caused analysis of the newly generated incident reports to be delayed by three to six months. This delay reduces the effectiveness of the ASRS as an early-warning system for decision-makers, aviation organizations, and government agencies.

The situation described above has become increasingly urgent in recent years due to the burgeoning growth of commercial air transportation, private aircraft, and unmanned aircraft systems in the aviation industry [1], thereby yielding a quickly mounting number of incident reports. Figure 1 shows annual incident reports ASRS received over the last 28 years. For instance, ASRS only received approximately 4,600 incident reports in 1981, compared with about 108,000 incident reports per month in 2019. Worse yet, the lack of timely and accurate analysis of the incident reports substantially reduces the value of the data, making effective safety prevention and improvement strategies increasingly challenging (Figure 1).

Safety in aviation transportation is crucial. Analyzing incident reports quickly and accurately on a large scale facilitates the decision-making process and makes early detection and prevention of potential hazards possible. In this study, we build a deep-learning model that can identify not only *primary factors* but also *contributing factors* with promising results described later on. The main contributions of our research to reduce gaps in extant research are summarized as follows:

Table 1: The example of an incident report and its analysis results.

Incident report submitted:
Narrative: busy session numerous over flights requiring course changes to avoid traffic. Six or seven arrivals to different airports descending through over flights and several departures. A satellite propeller arrival was coming in from the north at 10000 and an over flight was off the departure end NE bound at 11000 two F16 south departed routed north climbing to 15000 a military intercept was squawking 7777. I assumed this was in error and I informed the lead. I turned the aircraft to 020 heading to split the other traffic and allow the climb to continue. I consciously thought about re-assigning the altitude but after the squawking the turn and traffic issuance, I did not want to throw any more numbers at the pilot who would increase the transmissions and potential confusion. I wanted to get this guy on course and off my frequency but had to wait until he topped. The guy at 10000 made numerous other transmissions. Then looked at the F16 south and he was climbing very fast as I was about to transmit the F16 south showed 16000 and asked intermediate fix. They were cleared to the block. I said no, assigned altitude 15000, contact ZKC. No traffic observed, but at 560 knots aircraft can mess things up pretty quick.
Tabular data:

96 attributes:
- Time: 200905
- Local Time of Day: 1201 − 1800
- Relative Position: (missing)
- ⋮
- ⋮
- Cabin Light: (missing)

Results analyzed by human experts:
Primary Factor: human factors
Contributing Factors: human factors, procedure, aircraft
Synopsis: arbus flight crew landing on runway 8L at ATL reports a runway incursion after being instructed to taxi via delta; bravo; Victor; Foxtrot to the ramp. Crew failed to turn on to bravo and entered 8R at delta. An EMB170 crew had to reject their takeoff on runway 8R.

Originally, an incident report comprises two components, *Narrative* and *Tabular data*. In most cases, *Tabular data* is neither reliable nor useful because it is either missing or not quite related to the incident. After being reviewed by human experts, *Primary Factor*, *Contributing Factors*, and *Synopsis* are the conclusions generated from this incident.

Table 2: A comparison of our study with extant research related to aviation reporting system.

Studies	Research target	Data set	Algorithms	Performance
Tiller et al. [14]	Analyze close call incident reports to assess severity level	117 reports from the ASRS (2014–2016)	Bliss's taxonomy, a manual case-by-case review process	Modification on the close call taxonomy is needed, but results were not discussed quantitatively
Tanguy et al. [2]	Extract metadata and keywords from the narratives, and topic mining	86,912 qualified reports used from DGAC	N-Grams Support Vector Machine topic modeling	Incident reports classified to seven major topics, with about 78% F_1 score on average
Kuhn [15]	Automate the topic mining process	ASRS incidents from 2010 to 2015 (the exact number is not specified)	N-Grams topic modeling	Some incidents are closely related to key words, and topic modeling identified those well, but results were not evaluated quantitatively
Robinson [13]	Identify the contributing factors of the incident reports	7,484 incident reports from the ASRS	Latent semantic analysis	Identify the multiple factors of each incident; the accuracy needs significant improvement
Shi et al. [4]	Identify two primary causal factors of incidents with machine learning	168,227 incidents from the ASRS	Naive Bayes Hoeffding tree OzaBagADWIN	Automate to identify two most casual factors, and topic mining used to extract structured information
Our study	Identify the primary factor and multiple contributing factors of each incident from six most causal factors	181,651 incident reports from the ASRS	Deep recurrent neural networks	Demonstrate that deep learning is a powerful tool for processing complex textual data. We achieve best performance so far to identify the primary factor and contributing factors among related research

(1) Rather than directly addressing the task of classifying incident reports, we make an early attempt to introduce a well-trained deep-learning language baseline model that can "understand" general English texts, and then we refine our model based on the performance of the baseline model to cope with the incident reports. Our research shows that about 4% accuracy is gained.

(2) To the best of our knowledge, our study is the first attempt to perform a multiclass and multilabel operation on ASRS incident reports on a large scale. Our study pushes the application of deep learning methods in the safety management domain forward. We propose suitable metrics to evaluate the performance of this multiclass and multilabel classification, which is rarely used in extant research as they

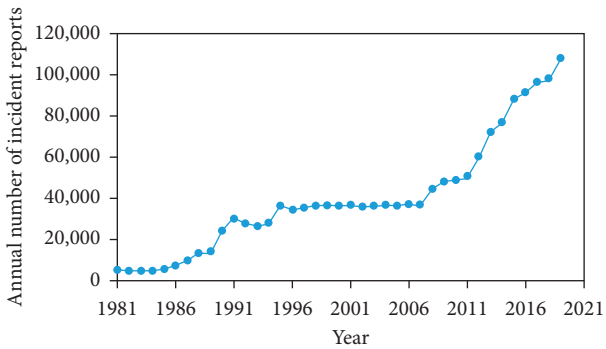


Figure 1: Annual incident reports ASRS received from 1981 to 2019.

primarily focus on binary or single-label classification.

(3) Our study demonstrates the high adaptability and reusability of deep learning methods. Therefore, our proposed deep learning methods are applicable to many tasks that demand text analysis, especially in an automated way. In addition, once the data is updated or the task is changed somewhat, the developed deep learning model can be modified accordingly without starting over from scratch.

This study establishes a fruitful research foundation for researchers who seek to apply deep learning methods to the solution of a myriad of text analysis problems in general and especially for those whose corpora include a customized vocabulary of technical terms. Our proposed approach sheds light on nontrivial optimizations to improve the baseline model's accuracy, as we strive to present a procedure to develop a deep learning model to help solve the pressing problem of aviation safety decision support.

The rest of this paper unfolds as follows. Section 2 is a review of relevant research. In Section 3, we describe the raw data and statistics and how to prepare them to be suitable for the training in the next step. Section 4 briefly introduces the main steps to build a deep recurrent network model using Python deep learning libraries and refine it based on our specific task. Section 5 epitomizes the experiments to determine hyperparameters in the model. We highlight the critical parameters that often significantly affect the performance of deep learning models, and we introduce new metrics to evaluate the results and compare them with related extant research. Section 6 discusses the potential implications of our research, and Section 7 presents the conclusions and limitations of this study.

2. Related Work

2.1. Automated Incident Analysis in Safety Management. Safety management is a continuous improvement process that reduces hazards and prevents incidents in aviation. The incident reporting system is a crucial part of safety management, as it collects data and evidence for decision-making, identifies potential risks to help prevent accidents, and provides examples to educate personnel. Extant research primarily concentrates on text mining techniques to automate the analysis of incident reports. Therefore, extant research has attempted to apply machine learning techniques to extract textual information. Table 2 compares this study with extant research that used aviation incident data. Tixier et al. [11] examined 2,200 construction incident reports by applying a rule-based automated content analysis system. The length of the sample reports presented in their paper was usually less than 50 words, and they primarily manually mapped keywords to specific incidents. Therefore, their proposed method is not easily applicable to lengthy and complicated narratives. Mousa et al. [12] proposed the XGboost algorithm to classify 13,165 highway-railroad crossing incidents and reported an accuracy of 99.11%. However, other baseline methods, such as Decision Tree or Random Forest, also achieve around 98.5% accuracy. Therefore, it is likely that the incident reports they were dealing with are naturally easy to differentiate. Shi et al. [4] applied manual feature engineering to the ASRS data set with Term Frequency-Inverse Document Frequency (TF-IDF) and fed the features into three supervised machine learning algorithms, Naive Bayes, Random Forest, and Support Vector Machine (SVM), to identify the two most frequent primary factors: "human factors" and "aircraft." The shortcomings of this research are that primary factors, "human factors," and "aircraft" combined account for about 81% of all incidents, and, even with only the two most frequent primary factors selected, the three traditional machine learning methods used in the research could only achieve an average accuracy about 81% at best. Therefore, a practical model that can handle more factors with improved accuracy is needed. Tanguy et al. [2] built classifiers with French national aviation occurrence data (DGAC[1]). The authors employed manual feature engineering using N-Grams and topic modeling and used the extracted features to train an SVM classifier. Rather than attempting to identify the primary factors from the incident reports, their goal was to discover the main topics of the incident, such as "cabin," "ground," and "weather." The disadvantage of their method is that, even when things like "cabin" and "weather" are mentioned in an incident report, they are not necessarily the actual factors that caused the incident. Robinson [13] was one of the first authors to tackle multilabel classification using an ASRS data set. The author built a latent semantic analysis (LSA) model, trained it with 4,497 incidents, and tested the model on 2,987 other incidents. However, the author reported poor model performance with an average F_1 score of 0.409 due to the small sample size used in the research overly ambitious attempt to classify all factors.

Our literature review indicates research gaps existing in the extant research. Most of the extant studies only use a relatively small number of data samples to develop their models. Models developed in this way may only be applicable to limited data sets. However, transportation incident reports are usually highly unstructured. Furthermore, although Shi et al. [4] used an extensive data set in their research, they only addressed the two most frequent factors, human factors and aircraft, which account for about 80% of all incidents, and ignored the rest. Such oversimplification

restricts the model to limited applications. The proposed methods in extant research are subject to two significant shortcomings: (1) a lack of high accuracy (less than 80%) and (2) a limited number of primary and contributing factors. Therefore, effectively automated identification of multiple incident factors to support decision-making remains one of the main challenges in aviation reporting systems. Due to various contributing factors such as human factors, aircraft, weather, and company policy [16], the inherent complexity of aviation operations requires reviewers with aviation experience to make sensible judgments. Accumulated evidence of the successful application of deep learning methods to the analysis of incident reports could bring about the acceptance of this approach as a solution to aviation safety management.

2.2. Emerging Deep Learning Methods in Transportation. In the last few years, deep recurrent networks, a subclass of deep learning methods, have been widely applied in transportation decision-making systems and have achieved promising results. Dong et al. [17] applied deep neural networks to predict traffic crashes. The study shows the advantages of deep learning methods over SVM, including automatic feature extraction, superior performance, and the ability to handle heterogeneous data. Cortez et al. [18] used bidirectional long short-term memory (LSTM) to predict emergency events using data from the Korean Ministry of the Interior in 2015, and the LSTM model showed better performance than SVM and time series models. A more recent aviation study [19] used recurrent networks to predict flight trajectory and their results illustrated the promising performance of the blended deep learning model in predicting flight trajectory and assessing en-route flight safety. Luo et al. [20] combined KNN and LSTM to predict traffic flow. KNN was used to address spatial data and LSTM for temporal data. The study reported that the deep learning method achieved superior performance on real traffic data. All the above studies have successfully shown the superiority of deep learning methods on large and unstructured data sets over traditional machine learning algorithms.

The deep neural network model, which combines the advantages of unsupervised and supervised learning algorithms, is superior to traditional machine learning algorithms in many respects, especially in this "Big Data" era. Instead of the manual feature engineering required by traditional machine learning algorithms, deep learning methods can extract intrinsic features without human intervention. The manual feature engineering is primarily based on word frequency statistics [21], such as TF-IDF and N-Grams. Its main shortcoming is that it has difficulty in capturing the relationships among textual data accurately. In deep neural networks, on the other side, the word is represented as a high-dimensional vector using a skip-gram technique [22]. In this way, intrinsic relationships among words and the meaning of each word can be constructed and calculated, and this approach has yielded outstanding results [23]. Second, another advantage of deep neural networks is that traditional machine learning methods primarily predict by merely counting the word frequencies or probabilities of words that appear together, rather than extracting the meaning of the word based on its semantic context. However, deep neural networks have the ability to "remember" or store previous information. This ability is beneficial for building relationships among words that do not appear close to each other. This ability is crucial to our tasks because incident reports may not be written in an organized and concise way. That is one of the main reasons why the automatic analysis of incident reports is challenging. Last, deep neural networks are naturally suitable for use with a large amount of textual data. More data is helpful to refine the word embeddings [24]. Word embeddings are also called word vectors. They are a way of converting textual data *o* numbers. Unlike other common ways of embedding, such as frequency embedding, TF-IDF, Count Vectors, and word vectors are initialized randomly, then trained, and refined with a large corpus of texts. The essence of word embedding is that all the other words in the context decide the value of a word vector. Mikolov et al. [25] developed this method, and it has gained significant attention in natural language processing since then. With word embeddings applied, the model can evolve along with the accumulation of incident reports, as the ASRS is constantly receiving them.

Despite being powerful and efficient type of algorithms successfully applied to many domains, deep learning methods have found limited implementation in transportation incident reporting systems, which require natural language processing. The goal of this paper is to cover this research gap by building deep recurrent neural networks that can automate aviation incident report analysis with better performance than extant research.

3. Data Preparation

3.1. Data Descriptive. We downloaded about 200,000 incident reports from the ASRS database ranging from January 1988 to July 2020 when accessed on October 2, 2020, yielding a total of 181,651 qualified reports. Other unqualified reports, such as those without labels or those that are too short (fewer than 20 words), are discarded. Every incident report is composed of four pieces of text from two persons (their narratives and callbacks), which we have combined as a single narrative text sent to our model. Figure 2 shows the distribution of the number of words and sentences in our data sets. The considerable variations of number of words and sentences make it more difficult to build a robust model.

There are 16 primary factors identified by human experts in aviation incidents; however, we only use incident reports involving the six most frequent categories of *human factors* (HF), *aircraft* (AC), *company policy* (CP), *procedure* (PR), *weather* (WE), and *airport* (AP), which make up 95% of the incident reports. Incidents attributed to rare factors are not considered in this research, because they only account for a fraction of all incidents and would need more data to generate meaningful results. We believe that our research thus achieves a reasonable balance in terms of performance, feasibility, and reasonable simplification. Table 3 lists all primary factors and their percentages of all incidents. The highlighted factors are used in this study and other rare factors are ignored.

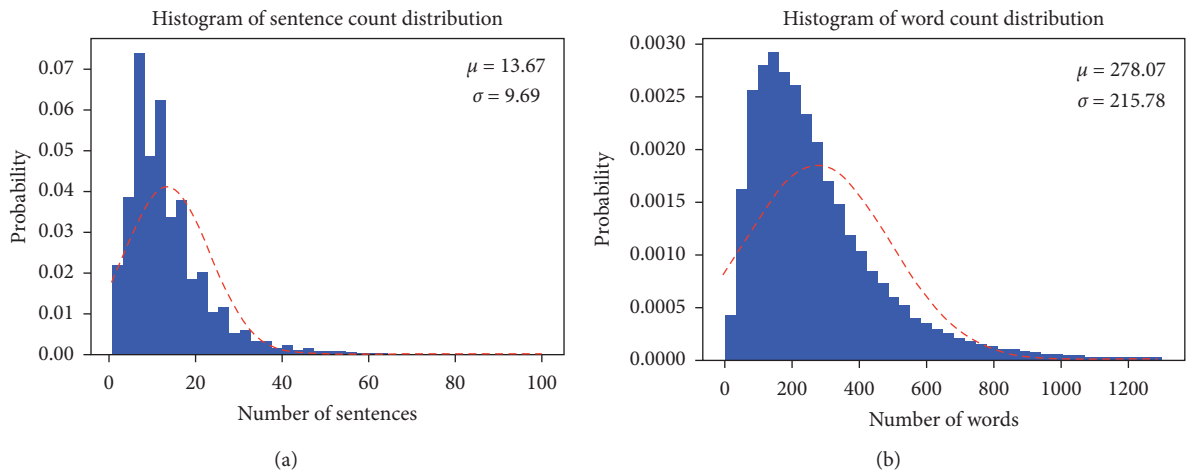


Figure 2: Distributions of lengths of incident reports based on the number of sentences and words.

Table 3: There are sixteen primary causal factors identified by the human experts in the database.

Primary casual factor	Count	Percentage in all incidents
Human factors	112,305	58.6%
Aircraft	43,119	22.5%
Company policy	7,676	4.0%
Procedure	7,626	4.0%
Weather	6,450	3.4%
Airport	4,475	2.3%
ATC equipment/buildings	2,803	1.5%
Chart or publication	2,519	1.3%
Environment, non-weather-related	2,180	1.1%
Airspace structure	1,163	0.6%
Equipment/tooling	465	0.2%
Manuals	338	0.2%
Staffing	238	0.1%
MEL	211	0.1%
Incorrect/unavailable part	154	0.1%
Logbook entry	32	0

The distribution of causal factors is highly unbalanced. Extant research primarily focus on the identification of the first two factors and ignore others. This study addresses the six most frequent factors, which account for as much as 95% of all incidents. Therefore, our solution is more applicable and feasible, because it can handle more factors, and is not targeting all factors, which causes the prediction performance to be worse due to the data unbalance.

In this research, we use narrative texts as the input to our model and, according to the input, our model predicts the primary (single-label) and contributing factors (multilabel) and compares them with the actual labels to evaluate the model's performance. We do not use the "Synopsis" section of each report as an extra input, because it is not the original content of the incident report and would make our automated text analysis less convincing.

Table 4 summarizes the essential statistics about multiple causal factors in ASRS data sets. Factor (or label) cardinality [26, 27] indicates that there are 1.47 factors (1 primary and 0.47 contributing factors) per report on average across all incident reports. This is the underlying reason for our decision to train our model to predict up to two factors for a single incident report, as mentioned in section 2. Identifying more than two factors for each incident report is not necessary in our research because cases of more than two factors are rare, and it would introduce unnecessary complexity without obvious performance gain. There are 28 distinct causal factor sets cooccurring in all incident reports, of which the most frequent combination is that of human factors and aircraft.

Table 5 shows the distribution of the six most frequent causal factors in detail. The overall occurrence of *human factors* (HF) is over 26 times more than that of *airport* (AP). The imbalance of the data distribution is likely to cause the classifiers to be biased toward the dominant category, in this case, human factors. Oversampling is applied to augment rare samples to overcome this issue. The other method we use to mitigate the bias is to apply a confidence threshold to *human factors*. Both are discussed in Section 5.

3.2. Data Preprocessing. We preprocess the narrative texts to reduce complexity and make the model more robust. Initially, the words in the report are tokenized into a list of its constituent words. Punctuation and stop words are removed in this step as they are not useful for text analysis [28]. Stemming and lemmatization are also applied to the input to decrease the number of distinct words and consequently reduce the model's complexity. To perform stemming and lemmatization accurately, a recognized Python library, the Natural Language Toolkit (NLTK) [29], is utilized. The ASRS extensively uses 537 acronyms for the words and phrases that frequently appear in narratives to make raw texts concise. For example, "STOL" stands for "Short Takeoff and Landing," and "VLF" represents "Very Low Frequency." These acronyms are decoded to their full words as the word vectors of acronyms are not seen in the pretrained word embeddings, which has been trained with the Wikipedia corpus. In addition, there are many meaningless words (or noise) existing in the corpus, such as "eeegl3," "shedcb," and "sewart." Thus, we remove any word that appears fewer than four times in our ASRS data sets. The study

Table 4: Important statistics about the utilized ASRS data set.

Multilabel statistics	Value
Number of utilized factors	6
Number of valid samples	172,990
Factor cardinality	1.47
Factor density	0.245
Number of distinct label sets	28
Most frequent label set	{Human factor, aircraft}

After cleaning and preprocessing, we use the six most frequent labels from 172,990 reports. On average, every report has 1.47 labels (label density of 0.245).

Table 5: The distribution of the number of labels along with distribution of labels within each number.

Number of labels	Total (%)	HF (%)	AC (%)	CP (%)	PR (%)	WE (%)	AP (%)
One	65.3	42.7	17.4	0.9	2	0.8	1.3
Two	24.3	14.6	4.6	2.01	1.05	1.16	0.8
Three	8.7	4.88	1.38	0.73	0.47	1.01	0.25
Four	1.6	0.8	0.2	0.15	0.08	0.3	0.08
Overall	100	63.0	23.6	3.8	3.6	3.3	2.4

The data is interpreted in this way; take the highlighted number for instance; 14.6% of all incident reports are marked exactly two causal factors (labels), and one of them is *HF*. It shows that *HF* prevails in both single and multiple labels.

[30] also used this straightforward but effective method to remove uncommon and useless words. In this way, many uncommon words are removed, while the important information of each incident report is kept intact. After preprocessing, a total of 6,960 unique words remain from 181,651 incident reports in this study.

As shown in Table 5, the distribution of the incident categories is highly imbalanced. Oversampling is used to augment the original data, because removing data from overrepresented classes, called undersampling, would not have been conducive to our deep learning approach, as deep learning improves with more data. Oversampling is a process that augments the data samples of underrepresented classes by copying them a certain number of times. In this study, incident reports labeled "aircraft" are copied two times, and those labeled "airport" ten times, and they are put back in the training data set. Finally, as shown in Table 6, of 181,651 incident reports, 80% are randomly picked as the training data set, 10% are used as the validation data set, and 10% are reserved as test data to measure model performance [31]. We apply oversampling after splitting the data to avoid data leakage between training, validation, and test sets. Unlike the validation data used by the model to monitor its performance during the training process, test data is kept isolated until the evaluation stage to guarantee the validity of the test data sets.

In this study, we only use oversampling to augment training data sets to identify primary factors. Regarding contributing factors, there is no noticeable performance gain from oversampling according to our experiments, because contributing factors are already mixed.

4. Methodology

4.1. Analysis and Processing of Aviation Incident Reports. The aviation incident reports are primarily free-form text describing each incident. A few incident reports may include some tabular data, such as the time and location, but the tabular data is missing in most incident reports. Therefore, the incident data has a strong temporal and spatial correlation because natural language is sequential, as the meaning of a word depends on the words that precede or follow it. However, traditional machine learning treats data (words) independently distributed in the context by following certain patterns that can be found statistically. Hochreiter and Schmidhuber proposed the first LSTM model [32], which is an advanced form of recurrent neural network (RNN), as it introduces "memory" and "forget" cells. These cells can effectively resolve problems such as vanishing gradient and long-term dependence with which RNNs struggle. This study uses an LSTM neural network model to process word vectors and make classifications.

The overall procedure of our model is shown in Figure 3. As mentioned in Section 1, we approach the problem by developing models that can identify the primary and contributing factors of the ASRS incident reports based on deep recurrent neural networks. Specifically, we start with a general unsupervised language model called Universal Language Model Fine-Tuning (ULMFiT), thoroughly trained by Wikipedia articles [33]. Next, we use an inductive transfer learning technique to refine this general model on our specific ASRS data sets to get familiar with the structure and semantics of the narrative text in the incident reports. Inspired by [34], we implement a universal language model based on Averaged Stochastic Gradient Descent Weight-Dropped LSTM (AWD-LSTM), a state-of-the-art variant of RNNs for language modeling and text classification tasks. The model uses a variety of effective regularization techniques that significantly improve the generalization performance of vanilla LSTM recurrent neural networks. Afterward, using supervised learning and 80% of the incident reports as training data sets, we build and fine-tune classifiers using the AWD-LSTM model and additional concatenation and feed-forward layers to predict primary and multiple contributing factors in the textual reports.

Table 6: The summary of the incident reports and their label distribution in the training set before and after data oversampling, as well as validation and test sets.

	Original	Train (oversampled)	Validation	Test
Human factors (HF)	87356 (62.8%)	87356 (25.4%)	10941 (64.0%)	16145 (63.4%)
Aircraft (AC)	32690 (23.5%)	65380 (19.0%)	3823 (22.4%)	6620 (26.0%)
Company policy (CP)	5335 (3.8%)	53350 (15.5%)	635 (3.7%)	1047 (4.1%)
Procedure (PR)	5321 (3.8%)	53210 (15.4%)	645 (3.7%)	1004 (4.0%)
Weather (WE)	4979 (3.6%)	49790 (14.5%)	623 (3.7%)	952 (3.7%)
Airport (AP)	3424 (2.5%)	34240 (10.0%)	428 (2.4%)	643 (2.5%)
Total	139105 (100%)	343326 (100%)	17095 (100%)	25451 (100%)

Validation and test data are maintained as imbalanced as the original training set to truly represent the data sample distribution.

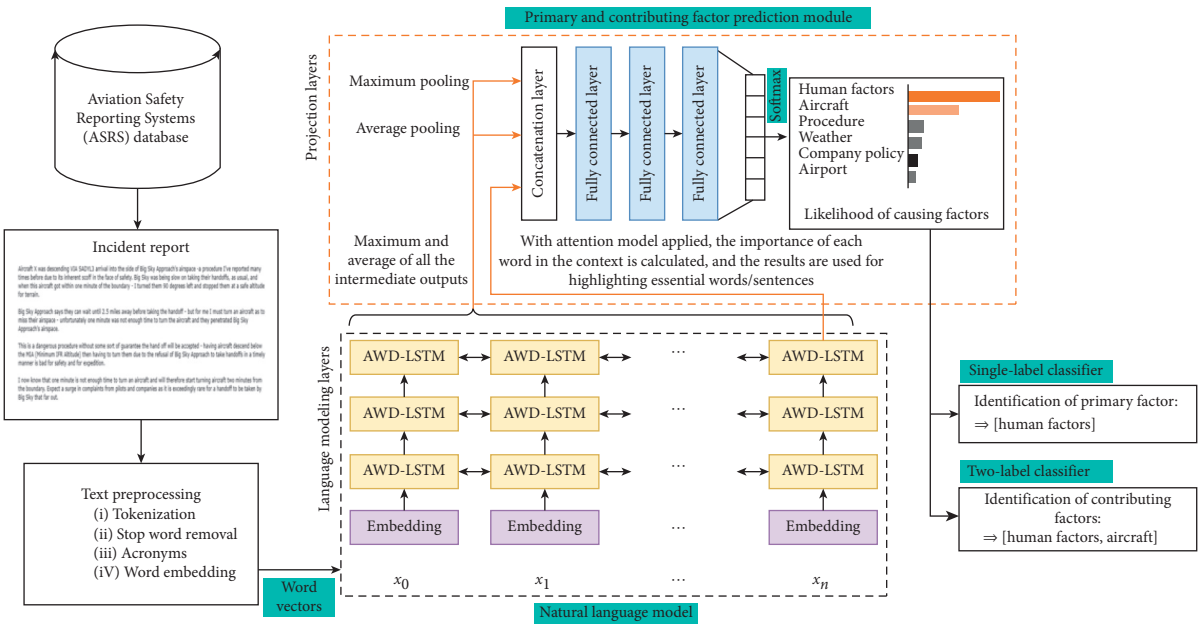


Figure 3: End-to-end diagram of the identification of factors of incident reports. Incident reports including the narratives are downloaded from the ASRS database. After being preprocessed, they are fed to deep neural network model which is composed of two components: (i) a language modeling module, an input layer of embedding and three stacked layers of bidirectional AWD-LSTM recurrent neural networks; and (ii) a prediction module, a flatten layer and three fully connected layers. After processing by these two modules, a probability score is assigned to each factor. Finally, the primary and contributing factors are predicted based on the ranking of probability scores.

We address the identification of the primary factors (single-label) and contributing factors (multilabel) as two different classification tasks, although they share the same architecture until the last layer. It might be tempting to use highest and second-highest probability factors as multilabel results, so that only one model is sufficient to classify multilabel, multiclass tasks. However, the experiment from this study shows inferior results with this approach, as the results are likely to be biased toward dominant factors in the data set. Instead, the training processes for single label and multiple labels have to run separately with corresponding truth labels. Table 3 shows a complete procedure of our approach. After the data preprocessing stage explicitly explained in Section 3, we apply deep neural networks on the textual data. The major steps are explained as follows.

4.2. The Baseline Natural Language Model. Unlike extant research, which does not use any textual data aside from the data used for the primary task of each study and thus restricts the quality and quantity of the data set, we first introduce a universal language model [35] that is pretrained with a large, well-prepared Wikipedia text corpus, thanks to Salesforce Research[2]. The benefits of this approach are threefold: (1) The pretrained open-source model is trained thoroughly. It is called "universal" as it covers a large set of textual data, including most of the words that appear in the incident reports. (2) The amount of available textual data is greatly increased. Even though we have 181,651 incident reports with a total of about 46 million words, this is still not a large enough corpus to train a deep neural network model well. Google[3] recommends a corpus of about 0.8 billion words. (3) This approach saves significant computational resources. Otherwise, a supercomputer would take one month to train a well-prepared language model, which is not feasible for most academic researchers.

4.3. Baseline Language Model Fine-Tuning. We have a well-made baseline natural language model, but the problem is that it seems to be unrelated to our specific task. After all, the

incident narrative data is different from the Wikipedia text corpus. This is where fine-tuning comes into play [36]. To make the baseline language model suited to our specific task, we refine our universal language model using the ASRS data set. Inspired by [34], we implement a universal language model based on AWD-LSTM.

4.4. Prediction of Primary and Contributing Factors. As Figure 3 shows, after the words have been processed by the language model, they are now presented in high-dimensional vectors and fed to artificial neural networks (ANNs) to generate the prediction. Extant research has proven ANNs to be successful at classification tasks [37]. Naturally, the one having the highest probability score among the six factors should be identified as the primary factor. However, due to the imbalance of the sample data and the narrative texts' intrinsic complexity, we apply novel adjustable thresholds to "human factors" only to control the rate of false positives, as discussed in more detail in Section 5. No threshold is applied to other primary factors or when identifying multiple contributing factors. In this way, we achieve a good balance among the six most common primary factors in the overall performance without adding too much complexity.

5. Experimental Setup and Result Discussion

As shown in Table 4, each report contains one primary factor and an average of 1.47 contributing factors. Therefore, we design the model to predict up to two contributing factors for each incident report after weighing the advantages and disadvantages of additional complexity. In this study, two classifiers are developed: (i) a *single-label classifier* to predict the primary factor and (ii) a *multilabel classifier* to predict up to two contributing factors. These two classifiers follow the same methodology explained in Section 4, except that different truth labels and label sets are used during the training step. This is a clear example of the adaptability and reusability of deep learning models. Usually, only the project layers need updates when the task is changed, while the main model remains the same. We will discuss the details of our experimental setup and results later in this section.

5.1. Configuration. In this section, we briefly discuss the configuration and critical hyperparameters of our model, that is, learning rate, batch size, hidden layer size, dropout, and so forth. We use a grid search algorithm [38] to find the optimal values that lead to the highest performance on the training set.

Both primary and contributing identification classifiers use a three-layer LSTM[4] model with 1152 hidden units in the hidden layer. We train our model on a Tesla V100-SXM2 GPU machine with 16 GB of memory. We use a batch size of 128 as optimum, based on the computing stability of the stochastic gradient descent and memory restrictions of the GPU machine. Each word is vectorized to 400 dimensions using a vocabulary size of 60,000. The optimal number of dimensions is often between 300 and 500, according to industry experiments and research [39]. In this study, the maximum length of a sequence is set to 700 words to avoid the diminishing returns of larger networks [40]. As shown in Figure 2, most of the incident reports have no more than 700 words; for reports having more words, all words beyond 700 are simply truncated and ignored. Thus, the input shape is (128, 700, 400).

As mentioned in Section 4, the deep RNN language model is based on the AWD-LSTM, which uses dropouts on the recurrent weights for effective regularization and prevents the model from overfitting. As a means of regularization, such dropouts can effectively reduce the overfitting problem [41]. In this study, the dropout values for the embedding, input/output of every intermediate layer, the output of the final layer, and the hidden-to-hidden weights (recurrent weight-dropped) are 0.25, 0.15, 0.1, and 0.2, respectively.

To train our deep neural network's parameters with ASRS incident reports, we use Slanted Triangular Learning Rate [33]. It quickly increases within the first few hundred iterations and then gradually decays until the epoch ends. This dynamic learning rate enables the model to learn quickly when the loss is high in the beginning and to gradually refine the parameters when the loss becomes smaller[5].

5.2. Retraining Effect on Language Modeling and Factor Identification. As mentioned in Subsection 4.3, AWD-LSTM, initially trained on a well-prepared wiki text corpus, is our baseline LSTM model. It is retrained using the ASRS data set to make it work well in this study. Such retraining is especially useful if the text data of the target task is massive. Figure 4 shows how the training loss, validation loss, and prediction accuracy of the language model change during the training epochs. Each epoch takes about 45 minutes to complete. Initially, the training loss and validation loss are reduced, and the accuracy gradually improves, which indicates that the model can make better predictions in each epoch. In other words, the model is learning. After certain epochs, in our case, after the 15th epoch, training loss continues to decrease linearly, while validation loss and accuracy stabilize at certain values, indicating the optimal time to terminate training; otherwise, the model will overfit on the training set, a notorious problem in deep learning [42]. In our study, retraining the language model improves the identification accuracy of the primary factor by 3.6%, consistent with the retraining gain described in the literature [33, 43].

5.3. Evaluation Metrics. Primary factor identification results are normalized to prevent the results of dominant classes from weighing too much. Therefore, in this study, percentages of true positives, false positives, and false negatives, rather than their counts, are used to calculate the precision and recall. Normalization puts more weight on rare classes, and this is usually more reasonable to measure classes that are not evenly distributed [44].

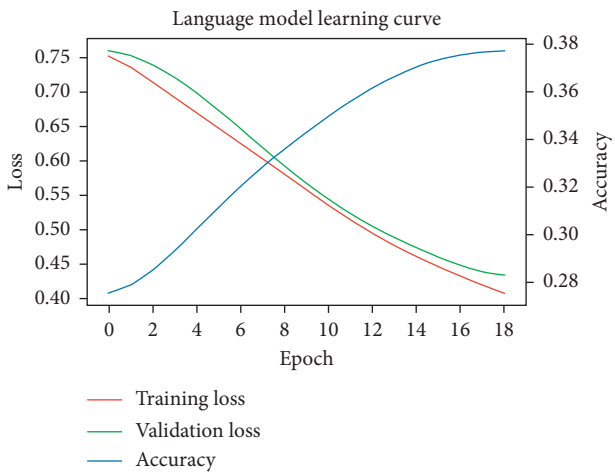


Figure 4: Language model learning curve. Accuracy here is defined as the percentage of predicted correct next word from a given vocabulary. Initially, the language model AWD-LSTM can achieve an accuracy of only 0.28; after 15 epochs, the accuracy improved to 0.38, a significant boost.

An "exact match" metric makes sense to evaluate the performance of the primary causal factor identification, as there is only one primary factor for each incident report. However, "exact match" does not work very well for evaluating the performance of multiple causal factor identification, because "exact match" completely ignores partial correctness. Thus, [45] introduces 11 common evaluation metrics for multiple causal factor (multilabel) identification. In this paper, hamming loss, micro-F_1, and macro-F_1 are selected to measure our results, as these three are commonly recognized and chosen in previous research [13, 46].

Hamming loss is the fraction of labels that are incorrectly predicted. Unlike "exact match," hamming loss is more forgiving in that it penalizes only the individual labels that do not match the truth labels [47]. Hamming loss is a loss function; thus the lower, the better.

Besides the hamming-loss metric, macro-F_1 and micro-F_1 are two conventional methods to evaluate the performance of multiple causal factor identification [48]. The critical distinction between macro-F_1 and micro-F_1 is that macro is an average per category, while micro is an average per sample point. These metrics are computed according to the following equations:

$$\text{Hamming loss} = -\frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}\left[h_{ij} \neq y_{ij}\right], \tag{1}$$

$$\text{macro } F_1 = \frac{1}{l}\sum_{j=1}^{l}\frac{2\sum_{i=1}^{m} y_{ij}h_{ij}}{\sum_{i=1}^{m} y_{ij} + \sum_{i=1}^{m} h_{ij}}, \tag{2}$$

$$\text{micro } F_1 = \frac{2\sum_{j=1}^{l}\sum_{i=1}^{m} y_{ij}h_{ij}}{\sum_{j=1}^{l}\sum_{i=1}^{m} y_{ij} + \sum_{j=1}^{l}\sum_{i=1}^{m} h_{ij}}, \tag{3}$$

where h_{ij} is the target, y_{ij} is the prediction, m is the number of samples, and l is the number of labels.

5.4. Primary Factor (Single-Label) Identification Performance. As "human factors" still account for 25.4% of all incidents after oversampling, the classifier tends to be biased toward "human factors." To further reduce the bias, we apply a confidence threshold to control the percentage of false positives in the "human factors" category. For example, a confidence threshold equal to 0.55 means that the classifier only labels an incident with "human factors" if it has 55% confidence or more; otherwise, the category with the second-highest confidence, even it is lower than HF, is chosen. See Table 7 for an example.

Primary factor identification results are shown in Table 8. We apply the threshold to the "human factors" class only to reduce the rate of its false positives because it greatly outnumbers the other classes. Based on our experiments with different thresholds starting from 0.3 to 0.7 with increment of 0.05, we find that an HF threshold of 0.55 effectively reduces the rate of HF's false positives. Considering that the data samples of each factor are imbalanced, we believe that micro-F_1 is a better way to assess the model's performance because micro-F_1 is an average per sample point (see equation (3)). As shown in Table 9, the micro-F_1 scores of all classes except WE are improved (Tables 8 and 9).

5.5. Contributing Factors (Multilabel) Identification Performance. In this study, each incident's contributing factors are prepared by combining the original primary and contributing factors (if any) of the incidents. An example is shown in Table 10.

As mentioned in Section 5, our model is designed to predict up to two factors for each incident report. Consequently, any prediction is definitely a mismatch for incidents that are labeled with more than two factors. Nevertheless, multilabel evaluation metrics consider partial match (see equations (1)-(3) in Section 5.3). Table 11 summarizes the multilabel performance of our model by each category and overall performance. Our model achieves an F_1 score of 0.763 by averaging four averages: micro-avg, macro-avg, weighted-avg, and sample-avg. As shown in Table 5, "human factors" and "aircraft" significantly outnumber the other four categories combined. Therefore, micro-avg, calculated by counting true positives, false negatives, and positives globally, is preferable for evaluating our model's performance. Sample-avg, average based on samples, and weighted-avg, average based on labels, are adjusted versions of micro-avg and output similar results. On the other hand, the macro-avg metric can be expected to generate the worst F_1 score as it treats all classes equally, totally ignoring the number of samples in each class. Thus, it is less accurate than the other three metrics due to data imbalance (Table 11).

5.6. Comparison of Our Results to Previous Studies. To better understand our model's performance, we compare our results with previous studies addressing similar tasks, as well as with a base model without fine-tuning. To make the comparison valid and convincing, we use the same data sets as the previous studies. Because single-label and multilabel tasks have different evaluation metrics, we compare them separately.

Table 7: An example of how the "HF threshold" affects the identification result.

	Probability of each factor						
HF threshold	HF	AC	CP	PR	WE	AP	Identification
Threshold = 0	0.42	0.37	0.09	0.03	0.03	0.06	HF
Threshold = 0.55	0.42	0.37	0.09	0.03	0.03	0.06	AC

If an HF threshold is specified, HF will only be identified when its probability exceeds the specified value; otherwise, the factor with the second-highest confidence is chosen. In this way, the bias toward the dominant factors is well compensated by tuning the threshold. Threshold = 0 (no threshold). Threshold = 0.55.

Table 8: Comparison of the confusion matrix of the single label with and without the threshold (orthogonal values highlighted).

		Predicted label					
Threshold = 0							
		HF	AC	PR	WE	CP	AP
Truth label	HF	0.92	0.05	0.01	0.01	0.01	0.01
	AC	0.16	0.82	0	0.01	0.01	0
	PR	0.57	0.05	0.33	0.02	0.03	0
	WE	0.33	0.07	0.01	0.59	0.01	0
	CP	0.51	0.13	0.03	0.02	0.31	0
	AP	0.51	0.07	0	0.02	0.04	0.35
Threshold = 0.55							
Truth label	HF	0.84	0.07	0.03	0.02	0.02	0.02
	AC	0.08	0.89	0.01	0.01	0.01	0.01
	PR	0.37	0.09	0.47	0.02	0.03	0.02
	WE	0.30	0.05	0.03	0.59	0.01	0.02
	CP	0.32	0.16	0.04	0.02	0.42	0.04
	AP	0.34	0.08	0.02	0.03	0.05	0.47

By applying a proper threshold, the model's ability to identify other rarer classes is significantly improved, and overall performance of HF is improved as well.

Table 9: After applying the threshold, the model's overall performance in terms of micro-F_1 score is improved, especially for rarer factors, as precision and recall become more balanced.

	Probability threshold = 0			Probability threshold = 0.55			F_1 score improvement
	Precision	Recall	Micro-F_1	Precision	Recall	Micro-F_1	Percentage
HF	0.306	0.92	0.502	0.373	0.84	0.516	+2.7%
AC	0.689	0.82	0.748	0.664	0.89	0.761	+1.7%
CP	0.756	0.31	0.440	0.778	0.42	0.545	+23.9%
PR	0.868	0.33	0.478	0.783	0.47	0.588	+23.0%
WE	0.882	0.59	0.706	0.855	0.59	0.702	− 0.5%
AP	0.971	0.35	0.514	0.83	0.47	0.596	+16.0%

Table 10: An example of how multiple labels are prepared for each incident report using one-hot encoding.

HF	AC	PR	WE	CP	AP	Truth label
✓	—	—	—	✓	—	[1 0 0 0 1 0]

1 indicates that a factor is present, and 0 indicates that a factor is absent. Matches and mismatches of multiple labels prepared in this way can be conveniently evaluated by the Python scikit-learn library [49].

Table 12 clearly shows that our model is superior to Shi et al.'s [4] in terms of label categories and model accuracy. We not only identify the six most common causal factors but also expand our model to address multiple causal factors. In addition, our HF accuracy is significantly better, while AC accuracy is equivalent. With the improved HF accuracy, the overall accuracy is improved significantly, as it is the most frequent class. Robinson's research [13] is the most closely related study we can find in terms of multilabel classification. He implements a latent semantic analysis algorithm to classify all 16 classes for only 4,497 incident reports, compared with our 138,392 reports for training. As mentioned in Section 1, the ten rarest classes account for less than 5% of total incident reports. Therefore, his research attempts to classify 16 classes with such little data are not very reasonable, and the result is inferior to ours. In addition, the advantages of the fine-tuned language model are also demonstrated, because it refines the word embeddings with the target data set. Table 12 indicates that the LSTM with the fine-tuned language model outperforms the one without fine-tuning by 3.3% on HF accuracy and 1.9% on AC accuracy in single-label classification. In multilabel

Table 11: A summary of our model's performance in identification of multiple causal factors.

	Precision	Recall	F_1 score
HF	0.88	0.93	0.90
AC	0.87	0.83	0.85
PR	0.70	0.46	0.56
WE	0.71	0.43	0.53
CP	0.65	0.37	0.47
AP	0.68	0.39	0.50
Micro-avg	0.84	0.77	0.80
Macro-avg	0.75	0.57	0.63
Weighted-avg	0.82	0.77	0.79
Sample-avg	0.88	0.84	0.83
Hamming loss = 0.091			

Table 12: A performance comparison of our method with previous research, regarding single-label and multilabel identification.

Studies	Algorithm	HF accuracy	AC accuracy	Remark
Shi et al. [4]	Naive Bayes	73.2%	81.1%	This study targets *HF* and *AC* only
	Hoeffding tree	74.9%	87.0%	
	OzaBagADWIN	76.5%	88.3%	
Our study	LSTM without fine-tuned language model	84.8%	85.1%	Our study achieves a better result regarding *HF* and can identify four more factors
	LSTM with fine-tuned language model	88.1%	87.0%	
Studies	**Algorithm**	**Hamming loss**	**F_1 score**	**Remark**
Robinson [13]	Latent semantic analysis	0.269	0.409	Impractically targeting 16 factors
Our study	LSTM without fine-tuned language model	0.135	0.628	Our study feasibly targets the six most frequent factors with promising results achieved
	LSTM with the fine-tuned language model	0.091	0.763	

The advantage of the deep learning methods over traditional machine learning methods is clearly shown.

classification, the LSTM with the fine-tuned language model has a lower hamming loss but higher F_1 score compared with the base model. To sum up, these results demonstrate that the use of a fine-tuned language model can improve classification accuracy.

6. Implications

We build two classifiers to identify the primary and contributing factors, using a deep recurrent network algorithm. These models are trained with the narrative texts of ASRS incident reports. With our classification models, the amount of incident report analysis done by human experts can be significantly reduced. When an incident report is generated, our first classifier identifies the primary factor and then properly indexes it into the database. Then, the second classifier identifies additional contributing factors. Our model can automate most of the tasks, and human experts may only need to check the incidents classified with low confidence by our model. The implications of our study are summarized in four perspectives presented below.

First, from the perspective of aviation safety reviewers, our study can help them facilitate the identification of causal factors. As demonstrated in Section 5, our model achieves an average accuracy of 82% on the six most common factors and about 89% on the two most common factors on average. In addition, our model has achieved the best multilabel, multiclass identification results compared with extant research. Our study has shown that this approach can identify causal factors for 95% of incident reports in the database with little human intervention. If they adopt our approach, aviation incident reporting systems can quickly issue initial results to relevant parties, such as air traffic controllers, airline companies, and airport authorities.

Second, incident reports that are identified with high confidence by our models do not require review by safety experts. Less than 4.7% of incident reports are predicted with low confidence (probability threshold $\leq$ 0.55). Safety experts may only need to review those incident reports to make sure causal factors are correctly identified. Figure 5 is an example of an incident report parsed by our model with an attention mechanism applied [50]. The attention mechanism is an algorithm to calculate each word and sentence's relative importance based on the required outputs. For instance, if the truth label (the output) is "aircraft," then words and sentences likely to be related to "aircraft" are assigned higher importance or probability in the incident texts. As Figure 5 shows, the highlighted words and sentences are likely the critical information associated with the true causal factors of the incident. These highlights can help safety experts locate

Busy session numerous over flights requiring course changes to avoid traffic. Six or seven arrivals to different airports descending through over flights and several departures. Satellite propeller arrival was coming in from the north at 10000 and an over flight was off the departure end NE bound at 11000 two F16 south departed routed north climbing to 15000 a military intercept was squawking 7777. I assumed this was in error and I informed the lead. I turned the aircraft to 020 heading to split the other traffic and allow the climb to continue. I consciously thought about re-assigning the altitude but after the squawking the turn and traffic issuance, I didn't want to throw any more numbers at the pilot who would increase the transmissions and potential confusion. I wanted to get this guy on course and off my frequency but had to wait until he topped. The guy at 10000 made numerous other transmissions. Then looked at the F16 south and he was climbing very fast as I was about to transmit the F16 south showed 16000 and asked intermediate fix. They were cleared to the block. I said no, assigned altitude 15000, contact ZKC. No traffic observed, but at 560 knots aircraft can mess things uppretty quick.

Figure 5: Example of narrative texts that have been processed by our model. With the attention mechanism application, the potential essential sentences or words are highlighted, and they are more comfortable for human experts to review.

the definitive information faster, which substantially expedites the manual review process. At the same time, safety experts' correct labeling of manually reviewed incident reports can improve the model's performance in the long run. This model can further evolve into a text summarization system by generating a "Synopsis" [51], which currently has to be generated by safety experts manually. By reviewing the "Synopsis" generated from each incident report, the number of incidents that a human expert can handle per unit of time is greatly increased.

Third, from the perspective of reporting systems, such automation makes the generation of statistical reports easier. Due to the voluntary nature of the reports submitted to ASRS, NASA mainly uses the data as a lower-bound estimate. For example, there were 112,305 human error incident reports submitted to the ASRS from January 1988 through July 2020. It can be confidently concluded that at least 112,305 human errors contributed to aviation incidents during this period. Based on this lower-bound estimate, decision-makers can determine whether a problem exists and requires further investigation [52]. It is easy to provide aggregated and even dynamic incident statistics once the causal factor identification is automated with satisfactory performance.

Fourth, the deep learning solution developed in this study, a very versatile technique, can be redesigned and adapted to different domains other than aviation. This study has chosen the ASRS as an explicit example to show how deep learning techniques can help safety experts process a large quantity of textual data quickly and accurately. The application of this technique can help aviation safety experts find emerging dangers and potential hazards promptly from a large volume of incident reports. Although the incident reports in other transportation domains might be different in terms of quantities, textual characteristics, report formats, and so forth, the methodology designed in this paper can be adapted to address those varied tasks.

7. Conclusion and Limitations

Incident report analysis is crucial to improve safety management in high-risk work environments. Though a large amount of incident data is generated every day with the advances in data storage management and Internet of Things (IoT), effective and timely utilization of these resources has been hampered by the tremendous human effort needed to identify incident causes. This study presents models that can automate causal factor identification of ASRS incident reports based on deep recurrent neural networks. Our results demonstrate that deep recurrent neural network algorithms, trained and fine-tuned with proper transfer learning techniques, are versatile enough to build classifiers to predict the primary factor or multiple factors with minor modifications. Therefore, an initial understanding of incident reports' factors can be gained from automated incident report analysis. Given these potential benefits, this study's promising results may encourage researchers to explore the application of deep learning algorithms to other domains, such as autotransportation, medical facilities, information technology failure, and injury reporting, where automated text analysis is much needed.

There are several limitations to this deep learning approach. Currently, we are only able to classify the six most frequent categories in ASRS data sets. Ten other much rarer categories, accounting for approximately 5% of all incident reports, are unaddressed, primarily due to the lack of sufficient sample data for training the deep learning approach. Additional efforts will be required to find a deep learning architecture that requires less data or to figure out effective ways to augment the limited data samples. Another limitation of our study is that we have limited our multilabel classifier to no more than two factors. However, about 9% of incident reports have more than two labels. A more sophisticated model may further improve identification accuracy. Finally, tabular data such as locations and time periods are not used in the deep learning model proposed in this study. Future studies can investigate the causal relationships between tabular data and incident factors to determine which locations or time periods are more likely to be associated with human factor-related incidents.

Data Availability

The data used in this paper was collected from asrs.arc.nasa.gov/search/database.html. Researchers can request the data from the ASRS, or they can download it from the website.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Trinity University's Faculty Research Start-up Fund and Summer Research Stipend Program for 2018.

References

[1] T. Ali, H. Khazaei, M. H. Y. Moghaddam, and Y. Hassan, *Machine Learning in Transportation*, Hindawi, London, UK, 2019.

[2] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal, "Natural language processing for aviation safety reports: from classification to interactive analysis," *Computers in Industry*, vol. 78, pp. 80–95, 2016.

[3] D. Harris and W.-C. Li, *Decision Making in Aviation*, Routledge, Oxfordshire, UK, 2017.

[4] D. Shi, J. Guan, J. Zurada, and A. Manikas, "A data-mining approach to identification of risk factors in safety management systems," *Journal of Management Information Systems*, vol. 34, no. 4, pp. 1054–1081, 2017.

[5] G. D. Edkins, "The indicate safety program: evaluation of a method to proactively improve airline safety performance," *Safety Science*, vol. 30, no. 3, pp. 275–295, 1998.

[6] C. Posse, B. Matzke, C. Anderson, A. Brothers, M. Matzke, and T. Ferryman, "Extracting information from narratives: an application to aviation safety reports," in *Proceedings of the 2005 IEEE Aerospace Conference*, pp. 3678–3690, Big sky, MT, USA, March 2005.

[7] R. S. Dodd, D. Eldredge, and S. J. Mangold, *A Review and Discussion of Flight Management System Incidents Reported to the Aviation Safety Reporting System*, The National Academies of Sciences, Engineering, and Medicine, Washington, DC, USA, 1992.

[8] J. Orasanu and L. Martin, "Errors in aviation decision making: a factor in accidents and incidents," in *Proceedings of the workshop on human error, safety, and systems development*, pp. 100–107, Citeseer, Glasgow, Scotland, 1998.

[9] M. Bao and S. Ding, "Individual-related factors and management-related factors in aviation maintenance," *Procedia Engineering*, vol. 80, pp. 293–302, 2014.

[10] T. Edwards, S. Sharples, J. R. Wilson, and B. Kirwan, "Factor interaction influences on human performance in air traffic control: the need for a multifactorial model," *Work*, vol. 41, pp. 159–166, 2012.

[11] J.-P. Antoine, B. Rajagopalan, and D. Bowman, "Automated content analysis for construction safety: a natural language processing system to extract precursors and outcomes from unstructured injury reports," *Automation in Construction*, vol. 62, pp. 45–56, 2016.

[12] S. Mousa, S. Soleimani, J. Codjoe, and M. Leitner, "A comprehensive railroad-highway grade crossing consolidation model: a machine learning approach," *Accident; Analysis and Prevention*, vol. 128, pp. 65–77, 2019.

[13] S. Robinson, "Multi-label classification of contributing causal factors in self-reported safety narratives," *Safety*, vol. 4, no. 3, p. 30, 2018.

[14] L. N. Tiller and J. P. Bliss, "Categorization of near-collision close calls reported to the aviation safety reporting system," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61, no. 1, pp. 1866–1870, 2017.

[15] K. D. Kuhn, "Using structural topic modeling to identify latent topics and trends in aviation incident reports," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 105–122, 2018.

[16] M. Abedin, V. Ng, and L. Khan, "Cause identification from aviation safety incident reports via weakly supervised semantic lexicon construction," *Journal of Artificial Intelligence Research*, vol. 38, no. 1, pp. 569–631, 2010.

[17] C. Dong, C. Shao, J. Li, and Z. Xiong, "An improved deep learning model for traffic crash prediction," *Journal of Advanced Transportation*, vol. 2018, Article ID 3869106, 13 pages, 2018.

[18] B. Cortez, B. Carrera, Y.-J. Kim, and J.-Y. Jung, "An architecture for emergency event prediction using lstm recurrent neural networks," *Expert Systems with Applications*, vol. 97, pp. 315–324, 2018.

[19] X. Zhang and S. Mahadevan, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decision Support Systems*, Article ID 113246, 2020.

[20] X. Luo, D. Li, Yu Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with knn and lstm," *Journal of Advanced Transportation*, vol. 2019, Article ID 4145353, 10 pages, 2019.

[21] C. D. Manning, "Probabilistic syntax," in *Probabilistic Linguistics*, MIT Press, Cambridge, MA, USA, 2003.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, Curran Associates Inc., New York, NY, USA, 2013.

[23] M. M. Najafabadi, F. O Villanustre, and T. M. Khoshgoftaar, "Naeem Seliya, and Randall Wald. Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, 2015.

[24] S. T Aroyehun and A. Gelbukh, "Aggression detection in social media: using deep neural networks, data augmentation, and pseudo labeling," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, NM, USA, 2018.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations*, Scottsdale, AZ, USA, 2013.

[26] F. C. Bernardini, B. Rodrigo da Silva, M. Rodrigo, and E. B. Mitacc Meza, "Cardinality and density measures and their influence to multi-label learning methods," *Learning and Nonlinear Models*, vol. 12, no. 1, pp. 53–71, 2014.

[27] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: a unified view and novel approaches," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 820–865, 2015.

[28] J. Nothman, H. Qin, and Y. Roman, "Stop word lists in free open-source software packages," in *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, Melbourne, Australia, 2018.

[29] E. Loper and H. Steven Bird, "Nltk: the natural language toolkit," in *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, Association for Computational Linguistics, Philadelphia, PA, USA, 2002.

[30] N. Mahmoudi, P. Docherty, and P. Moscato, "Deep neural networks understand investors better," *Decision Support Systems*, vol. 112, pp. 23–34, 2018.
[31] H. Moradi, W. Wang, A. Fernandez, and D. Zhu, "Upredict: a user-level profiler-based predictive framework for single Vm applications in multi-tenant clouds," 2019, http://arxiv.org/abs/1908.04491.
[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[33] J. Howard and S. Ruder, "Fine-tuned language models for text classification," 2018, https://arxiv.org/abs/1801.06146.
[34] S. Merity and R. Socher, "Regularizing and optimizing LSTM language models," 2017, https://arxiv.org/abs/1708.02182.
[35] E. Grave, J. Armand, and N. Usunier, "Improving neural language models with a continuous cache," 2016, https://arxiv.org/abs/1612.04426.
[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[37] C. Gleue, D. Eilers, H.-J. von Mettenheim, M. H. Breitner, and Breitner, "Decision support for the automotive industry," *Business & Information Systems Engineering*, vol. 61, no. 4, pp. 385–397, 2019.
[38] J. Y. Hesterman, L. Caucci, M. A. Kupinski, H. H. Barrett, and L. R. Furenlid, "Maximum-likelihood estimation with a contracting-grid search algorithm," *IEEE Transactions on Nuclear Science*, vol. 57, no. 3, pp. 1077–1084, 2010.
[39] Zi Yin and Y. Shen, "On the dimensionality of word embedding," in *Advances in Neural Information Processing Systems*Curran Associates Inc., New York, NY, USA, 2018.
[40] K. Greff, R. Srivastava, K. Jan, and B. R. Steunebrink, "Lstm: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, 2014.
[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, UK, 2016, http://www.deeplearningbook.org.
[43] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.
[44] B. Peter, "The normalized recall and related measures," *SIGIR Forum*, vol. 17, no. 4, pp. 122–128, 1983.
[45] X. Zhu, "A unified view of multi-label performance measures," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3780–3788, Sydney, Australia, 2017.
[46] P. Probst, Q. Au, G. Casalicchio, C. Stachl, and B. Bischl, "Multilabel classification with R package mlr," *The R Journal*, vol. 9, no. 1, pp. 352–369, 2017.
[47] G. Tsoumakas and I. Katakis, "Multi-label classification," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.
[48] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information Retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.
[49] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[50] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, 2016.
[51] M. Moradi, G. Dorffner, and M. Samwald, "Deep contextualized embeddings for quantifying the informative content in biomedical text summarization," *Computer Methods and Programs in Biomedicine*, vol. 184, Article ID 105117, 2020.
[52] "ASRS database statistics," 1994, https://asrs.arc.nasa.gov/publications/directline/dl8_stat.htm.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6117890, 11 pages
https://doi.org/10.1155/2021/6117890

Research Article

Enhancing Mixed Traffic Flow Safety via Connected and Autonomous Vehicle Trajectory Planning with a Reinforcement Learning Approach

Yanqiu Cheng,[1,2] Chenxi Chen,[2] Xianbiao Hu,[2] Kuanmin Chen,[1] Qing Tang,[2] and Yang Song[2]

[1]*Department of Traffic Engineering, College of Transportation Engineering, Chang'an University, Xi'an 710064, Shaanxi, China*
[2]*Department of Civil, Architectural and Environmental Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA*

Correspondence should be addressed to Xianbiao Hu; xbhu@mst.edu

Received 16 April 2021; Revised 23 May 2021; Accepted 4 June 2021; Published 14 June 2021

Academic Editor: Chi-Hua Chen

The longitudinal trajectory planning of connected and autonomous vehicle (CAV) has been widely studied in the literature to reduce travel time or fuel consumptions. The safety impact of CAV trajectory planning to the mixed traffic flow with both CAV and human-driven vehicle (HDV), however, is not well understood yet. This study presents a reinforcement learning modeling approach, named Monte Carlo tree search-based autonomous vehicle safety algorithm, or MCTS-AVS, to optimize the safety of mixed traffic flow, on a one-lane roadway with signalized intersection control. Crash potential index (CPI) is defined to quantitively measure the safety performance of the mixed traffic flow. The CAV trajectory planning problem is firstly formulated as an optimization model; then, the solution procedure based on reinforcement learning is proposed. The tree-expansion determination module and rollout termination module are developed to identify and reduce the unnecessary tree expansion, so as to train the model more efficiently towards the desired direction. The case study results showed that the proposed algorithm was able to reduce the CPI by 76.56%, when compared with a benchmark model without any intelligence, and 12.08%, when compared with another benchmark model that the team developed earlier. These results demonstrated the satisfactory performance of the proposed algorithm in enhancing the safety of the mixed traffic flow.

1. Introduction

Connected and automated vehicles (CAVs) have been demonstrated to have great potentials for future transportation systems [1–4]. Compared with human-driven vehicles (HDVs), CAVs behave accurately as they are controlled by the computer algorithms, and their trajectories can be adjusted with predefined intelligence to achieve objectives such as minimizing delays and/or fuel consumptions at roadway intersections. This process is named longitudinal trajectory planning and is an important task to realize the full potentials of CAVs. Data from on-board equipment (e.g., in-vehicle sensors, radar, camera, and lidar) and remote facilities (e.g., DSRC/Cellular, GNSS/IMU, and priori map) can be utilized to schedule CAV trajectory [5].

Plenty of studies on CAV longitudinal trajectory planning have been conducted. For example, Chen et al. [6] proposed a centralized control method for CAVs by using a cost function which included CAV safety, efficiency, and ride comfort as the minimization objective. The robust platooning was formulated as a Min-Max Model Predictive Control (MM-MPC) problem, where optimal accelerations were generated to minimize this cost function. Wu et al. [7] presented an optimal longitudinal control strategy for a homogeneous CAV platoon. A linear-quadratic optimal controller was designed considering a comprehensive

perspective, including driving safety, efficiency, and ride comfort, with three performance indicators including vehicle gap error, relative speed, and desired acceleration. Malikopoulos et al. [8] provided a decentralized theoretical framework for coordination of CAVs. Rear-end, speed-dependent safety constraint had been taken into account. Research studies with similar objectives can also be found in [9–15].

While significant progress on CAV longitudinal trajectory planning can be observed in the abovementioned literature, one thing that is largely missing is the impact of CAV longitudinal trajectory planning algorithms to the safety of traffic flow, and subsequently, how should we design CAV longitudinal trajectory planning algorithms to minimize the probability of crash occurrence. To clarify, in most abovementioned works, the objective of trajectory planning is usually to reduce travel time or fuel consumption, and CAV safety is usually built in the model as a constraint, rather than an objective. In addition, the consideration of driving safety is usually limited to the CAV itself, instead of the other HDVs in the traffic flow. However, learning from the driving safety and human behavior research, traffic crash happens most frequently when the vehicles are changing speed, e.g., accelerating or decelerating at intersections. In a mixed traffic flow environment with both CAV and HDV, the CAV control algorithm will not only impact the movement of the CAV, but, through traffic flow shockwave propagation, will also influence the driving behavior of the HDVs at upstream locations. As such, it should be noted that the safety impact of CAV is not only limited to the CAV itself but also to the surrounding HDVs as well, and a good longitudinal trajectory planning algorithm needs to consider all of these and aims to minimize the crash potential of the entire traffic flow.

Methodologically, CAV trajectory scheduling is still a sophisticated problem, considering the great challenges from the highly stochastic nature of human driving behaviors and almost infinite decision-making states in real-world mixed traffic context. One common and effective approach to simplify the above complicate problem is to divide a vehicle trajectory into several segments. In other words, vehicles are usually set to the same cruising speed, or with constant acceleration/deceleration, at each stage. For example, He et al. [16] proposed a multistage approximation control model to solve the optimal trajectory problem. First, the vehicle cruised at the speed calculated by their algorithm and then accelerated/decelerated to a final speed when passing through the intersection. Wu et al. [17] divided the whole vehicle control process into a sequence of control stages and each control stage was formulated as an individual optimal control problem involving spatial and temporal constraints induced by the presence of vehicle queues. In [18], the vehicle was supposed to accelerate to different optimal cruising speeds by few speed guidance which also divided the roadway. In [19], the roadway was separated into three segments by two individual variable speed limits (IVSL). After those IVSLs, vehicle speed was adjusted to a final constant value so that their trajectories are smooth. Similar method can be found in [20, 21], in which each vehicle trajectory was broken into a few sections to decompose the originally hard trajectory design problem to a simple one. Although the abovementioned approach does make the model analytically solvable and help reduce the computational burden, such assumptions sacrificed the modeling realism and were not flexible to account for the uncertainty of human driver behaviors in a mixed traffic environment.

Considering the modeling techniques of trajectory planning, the computational complexity and algorithm runtime are directly related to modeling realism and the market penetration rate (MPR) of CAV. One way to reduce the complexity of the model is to only consider the pure CAV traffic, i.e., a traffic environment without any HDVs. In fact, large amounts of research studies on CAV trajectory planning were under this assumption. For example, Lee and Park [22] developed a CVIC algorithm for manipulating individual automated vehicle into crossing an intersection without colliding with other vehicles in a 100% MPR AVs environment. Wang et al. [13] proposed a rolling horizon control framework to control all vehicles' trajectory, which were equipped with driver assistance systems by optimizing a cost function reflecting different control objectives. Under the same assumption, Ahn et al. [23] developed an eco-drive system that combines an eco-cruise control algorithm and state-of-the-art car-following models. Zhou et al. [15] proposed a reinforcement learning-based approach to train a CAV platoon to pass through the intersection with a steady speed. The same research context can be found in [24–29]. In the above research studies, although it was able to simplify the model and improve calculation efficiency under the pure CAV environment, the applicability of the models was greatly reduced.

To deal with the abovementioned issues, this study proposes a model-free trajectory planning approach for improving the safety of mixed traffic flow of HDV and CAV, named Monte Carlo tree search-based autonomous vehicle safety algorithm, or MCTS-AVS. We quantify the safety level of the mixed traffic flow by using crash potential index (CPI) as the minimization objectives. The CAV trajectory planning problem is firstly formulated as an optimization model, and then, a solution procedure based on reinforcement learning is proposed. The tree-expansion determination module and rollout termination module are developed to identify and reduce the unnecessary tree expansion, so as to train the model more efficiently towards the desired direction. These modeling efforts lead to the improvement of algorithm solution quality and safety performance. Finally, the proposed algorithm was implemented and tested in a one-lane roadway with signalized intersection control.

2. Notations

As a convenient reference, the mathematical notations used in this section are presented below.

t, T: discrete time step, and the time horizon

$s(t)$: state at time t

$a_{CAV}(t)$, $\mathscr{A}_{s(t)}$: action of CAV, and set of all actions at state s and time t

$y_i(t)$: the distance of vehicles i from roadway entrance, at time t

$Y(t)$: an array that stores vehicles' distance from roadway entrance, at time t

$v_i(t)$: the speed of vehicle i, at time t

$V(t)$: an array that stores vehicles' speed, at time t

$d_{hi}(t)$: the distance headway of vehicle i, at time t

$D(t)$: an array that stores vehicles' distance headway, at time t

t_g, t_y, t_r, t_c: durations of green, yellow, and red signals, and cycle length

Δt: the shortest time interval

v_s: speed limit

l_{seg}: length of the roadway segment

l_a: average vehicle length

3. Methodology

3.1. Model Formulation

3.1.1. Problem Setting and Decomposition. We believe that, in the near future, the mixed traffic flow that composes of multiple HDVs and CAVs traveling on arterial segment will be a general scenario, as opposed to pure CAV traffic flow. This is because transitioning to fully CAV traffic might be a time-consuming process. It also implies that we will have a mixture of CAV and HDV in the mixed traffic flow, and the traffic dynamics become complex. To simplify the CAV control problem, this mixed traffic flow is firstly decomposed into several "basic interactive unit (BIU)," as illustrated in Figure 1. After the decomposition, each CAV is involved in one BIU, and the rest of the vehicles in the platoon are HDVs. In the Figure 1, the number of HDVs might be one or multiple, or there might be no CAV at all. As such, the mixed traffic flow problem can be converted into a trajectory optimization problem for each BIU, which significantly reduced the total computational complexity.

There are two reasons for such decomposition. First, if an HDV is driving in front of a CAV, due to the human nature, it will drive according to speed limit or prevailing cruising speed, and as a result, its behavior is not impacted by the CAV behind it. Second, CAV is subjected to the speed limit or current traffic conditions, as it cannot drive faster than a typical HDV. On the contrary, when it slows down to a speed that is lower than HDV, it becomes a moving bottleneck, and all HDVs behind it are forced to slow down and follow this CAV. To summarize, for mixed traffic flow control problem, we will always have a CAV that is leading the platoon and potentially multiple HDVs behind the CAV, in each basic interactive unit. Such decomposition is also frequently used in the previous literatures.

3.1.2. State Transition. To describe state transition, we use $s(t) = \{Y(t), V(t)\}$ to represent the state of mixed traffic flow at time, where $Y(t) = \{y_i(t), \quad i = 1, 2, \ldots, k\}$, $V(t) = \{v_i(t), \quad i = 1, 2, \ldots, k\}$, and k is the total number of vehicles. Then, $v_i(t)$ is updated by $v_i(t+1) = v_i(t) + a_i(t)$. CAV moves with action $a_{\text{CAV}}(t) \in \mathscr{A}_{s(t)}$ at time t. For the HDVs in the traffic flow, there are two distinct scenarios: (1) when HDVs are relatively far away from the intersection, their behaviors are mostly car-following (CF) and can be described by the classic CF model; (2) when HDVs are getting close to the intersection, the vehicle behaviors are subject to the signal lights. In other words, vehicles will drive through the intersection when the light is green or if they cannot come to a safe stop when the yellow light is on. Otherwise, it will slowdown and stop before the stop line. The HDVs behavior of these two scenarios are illustrated in Figures 2(a) and 2(b), and both of them follow the vehicle constraints, including collision avoidance and speed limit, as well as vehicle kinematics.

To describe the velocity decision-making of the HDVs for the first scenario, the general GM model considering stochastic HDVs behavior is employed. Compared with the classic intelligent driver model (IDM) which was introduced in [30], the GM model has the following advantages. (1) Human perception reaction time, speed difference, and space headway were involved in this model as a simple structure. It enables HDV trajectories' simulation rapidly but without losing too much detail. (2) A random term to reveal the uncertain factors of human drivers behavior was also been considered. This makes the model closer to the real scenario and a higher applicability. The specific formulation of GM model is shown as

$$a_i(t) = \alpha * v_i(t)^{\beta} \frac{\Delta v_i(t - t_{\text{reaction}})}{\Delta x_i(t - t_{\text{reaction}})^{\gamma}} + \epsilon_i(t), \tag{1}$$

where $a_i(t)$ is the acceleration value of the human drive vehicle i at time t, $v_i(t)$ is the vehicle's speed, t_{reaction} denotes the human perception reaction time, $\Delta v_i(t - t_{\text{reaction}})$ is the speed difference between the target vehicle and its leading vehicle at time $(t - t_{\text{reaction}})$, $\Delta x_i(t - t_{\text{reaction}})$ is the space headway, α, β, and γ are the parameters to calibrate, and $\epsilon_i(t)$ is a random term associated with vehicle i at time t. Several researchers (e.g., [31]) calibrated these parameters with collected data in real world. After Δt, state transition $s(t) \longrightarrow^{\Delta} t\ s(t+1)$ is realized by

$$y_i(t+1) = y_i(t) + v_i(t) + \frac{1}{2} a_i(t) t^2, \quad i = 1, 2, \ldots, k. \tag{2}$$

3.1.3. Crash Potential Index Function. Considering the movement of the vehicles in the traffic, we divide the traffic flow states into two types to further evaluate the safety performance of the current state. In general, when the vehicle velocity is less than the rear vehicle, two vehicles tend to be close, and the traffic flow has potential crash risk. We define this kind of state as a crash potential state, as shown

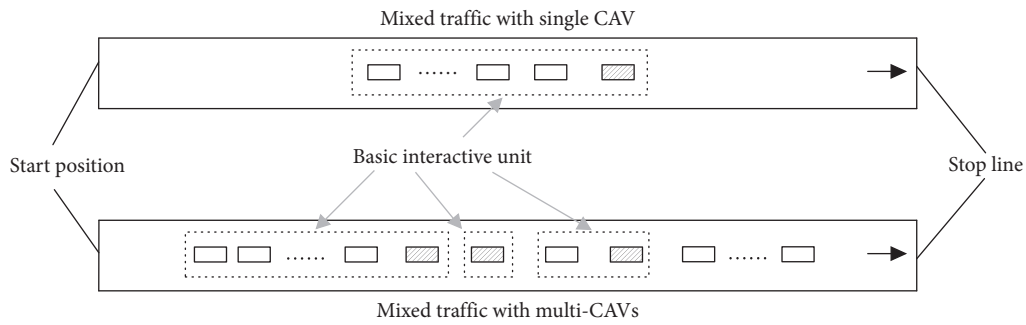


Figure 1: Decomposition of mixed traffic flow into basic interactive units.

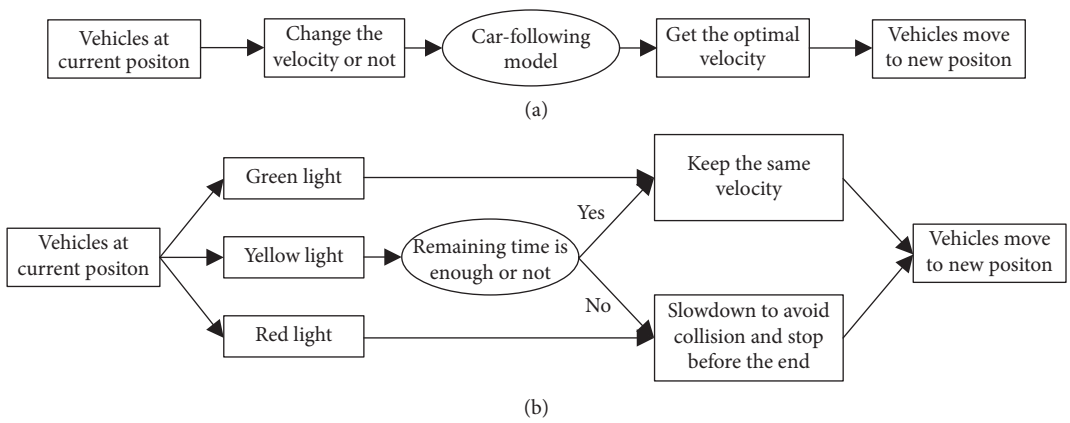


Figure 2: HDVs' behavior in roadway with signal control intersection. (a) HDVs are far from the signal light and (b) HDVs are close to the signal light.

on the left side of Figure 3. For example, when the signal light changes from green to yellow, the leading vehicle slows down and the traffic flow gets dense. On the contrary, when the vehicle velocity is greater than or equal to the rear vehicle, the distance headway will remain the same or increase, and there is less risk of collision in this traffic. This kind of state is defined as a safe state. For example, when the signal light changes from red to green, the leading vehicle begins to accelerate and the distance headway increases gradually, as shown on the right side of Figure 3.

To quantify the safety degree of a traffic flow, we defined a crash potential index (CPI) function as

$$X(t) = \begin{cases} \sum_{i=1}^{k} \left(v_{i+1}(t) - v_i(t)\right), & i = 1, 2, \ldots, k, \text{ if } v_i(t) < v_{i+1}(t) \\ 0, & \text{else,} \end{cases} \tag{3}$$

where $X(t)$ is the CPI value of this traffic flow at time t and k is total number of vehicles. The cumulative value considers the above two states: the speed difference between two adjacent vehicles is calculated when they are close to each other or zero when the two adjacent vehicles are far away or relatively slow. This value directly reflects the overall crash potential degree of the traffic flow.

3.1.4. Optimization Model. The overall optimization problem is represented by

$$\text{minimize} \sum_{t}^{T} X(t). \tag{4}$$

The feasible region for CAV action $a_{\text{CAV}}(t)$ at time t is subjected to

$$y_{\text{CAV}}(t+1) = y_{\text{CAV}}(t) + v_{\text{CAV}}(t) + \frac{1}{2} a_{\text{CAV}}(t) t^2, \quad t = 0, 1, \ldots, T, \tag{5a}$$

$$v_{\text{CAV}}(t+1) = v_{\text{CAV}}(t) + a_{\text{CAV}}(t), \quad t = 0, 1, \ldots, T, \tag{5b}$$

$$\left|a_{\text{CAV}}(t)\right| \le A, \quad t = 0, 1, \ldots, T, \tag{5c}$$

$$0 \le v_{\text{CAV}}(t) \le v_s, \quad t = 0, 1, \ldots, T, \tag{5d}$$

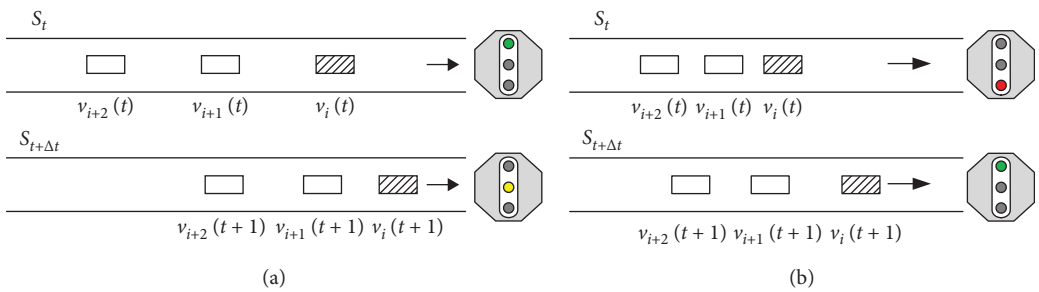


Figure 3: Two types state for a traffic flow: (a) crash potential state and (b) safe state.

where T is the time at the end of mixed traffic flow travel (e.g., get through an intersection). A is the upper limit of the absolute value of CAV acceleration.

3.2. Solution Algorithm

3.2.1. UCT Formulation. The problem in equations (4) and (5) is a challenging nonlinear program (NLP) with a huge state space, which makes the problem computationally intractable. This is because, at a given time t, the state of this problem is defined by a list of specific input features to describe the current system status and is required for any reinforcement learning algorithm. For a mixed traffic flow, many variables can be used to describe the state, for example, vehicle's distance from roadway entrance, vehicle velocity, accelerations, spacing/time headways between vehicles, elapsed time, and signal light color and their remaining duration. Obviously, when more features are selected, more details of the state will be captured. However, excessive number of state elements may directly lead to an exponential growth of the state space and lead to the "curse of dimensionality." As a result, a huge state space will come with a higher memory requirement and computational burden. Therefore, the features have to be chosen carefully.

In this study, we choose to use a combination of time, vehicle location, and vehicle speed to represent the time, in which the vehicle location and speed are two arrays that include information of all vehicles in the traffic flow. However, even with these 3 limited variables, once we discretize the time, space, and speed dimensions, this model becomes high-dimensional in state and is very challenging to solve and as such we have to rely on the reinforcement learning approach. In this study, we developed a heuristic algorithm, Monte Carlo tree search-based autonomous vehicle safety algorithm, or MCTS-AVS, to solve this problem by searching near-optimum action at every time step for CAV.

Typical MCTS algorithm consists of four steps: selection, expansion, simulation, and backpropagation [32, 33]. UCT algorithm (upper confidence bounds for trees) is employed to the first step of MCTS-AVS, as it can well balance the dilemma between exploration and exploitation part of a selection policy. The underlying mechanism for UCT, which is denoted by π_{UCT}, is described by the following formula:

$$\pi_{\mathrm{UCT}} = \arg\max\left\{Q''_{\mathrm{UCT}}(s,a) = Q_{\mathrm{UCT}}(s,a) + \mathbb{C}\sqrt{\frac{\ln(n(s))}{n(\mathrm{s},\mathrm{a})}},\ a \in \mathbb{A}\right\}, \tag{6}$$

where π_{UCT} is the selected policy, s is system state, a is action, $\mathbb{A}$ is the set for all actions, $n(s)$ is the total number of times a state s has been visited, $n(s,a)$ is the number of times action a has been selected in state s, $Q_{\mathrm{UCT}}(s,a)$ is the empirical cumulative reward, averaged over all iterations, when action a has been selected in state s, and $\mathbb{C}$ is a problem-dependent parameter to control the balance between exploitation and exploration. Equation (7) is defined to calculate the value of reward $Q(s, a)$:

$$Q(s,\ a) = \frac{\sum_{i=1}^{n(s,a)} X_i}{n(s,a)}, \tag{7}$$

where X_i denotes the reward of ith simulation associated with action a. The safety objective functions were modeled by equation (3). This objective is focused on the crash potential index. The expectation was that, by adjusting the movement of CAV, the crash potential of the mixed traffic flow can be reduced.

3.2.2. Tree-Expansion Determination Module. When CAV launches a general MCTS algorithm, it will run four steps at any time step. However, sometimes some operations were neither necessary nor helpful in improving the solution quality during the actual operation process. In other words, if the traffic condition was not much changed compared with the last moment, triggering of MCTS does not bring any new information to the simulation, but instead may introduce random noise and grow the tree towards an undesired direction. Additionally, such operation brings significant concerns to the algorithm run time and leads to a waste of memory and CPU resources.

To determine when should the tree expansion be prohibited, we analyze the "marginal impact" of a CAV movement. While CAV performs an action, the HDV that is immediately behind CAV would find a different time headway, and thus, its speed might be adjusted according to equation (8). To determine the degree of adjustment, we perform the partial derivative and can derive the acceleration/deceleration value as follows:

$$\dot{V}\left(d_{hi}^{t}\right) = \frac{\partial \mathrm{v}}{\partial \mathrm{d}} = 16.8 * \frac{\partial\left[\tanh 0.0860\left(d_{hi}(t) - 25\right) + 0.913\right]}{\partial \mathrm{d}} = 1.448\left(1 - \tanh^{2} 0.0860\left(d_{hi}(t) - 25\right)\right) d\left(d_{hi}(t)\right). \tag{8}$$

It should be noted that equation (8) merely quantifies the impact of CAV to the vehicle that follows immediately behind it. If multiple vehicles are following CAV, the impact would propagate to the upstream vehicles in the form of shockwave. As such, the total impact is the summation of all vehicles behind CAV, i.e.,

$$\dot{\mathrm{Sum}}(V) = \sum V\left(d_{hi}(t)\right) = \sum 1.448 * \left(1 - \tanh^{2} 0.0860\left(d_{hi}(t) - 25\right)\right) d\left(d_{hi}(t)\right), \tag{9}$$

and $\forall i$ behind the CAV vehicle.

3.2.3. Rollout Termination Module. In the simulation step, rapid rollout algorithm is employed to update $Q(s, a)$ value in equation (7) as follows. For a basic simulation, CAV moves with an action that is drawn randomly from the action set, until all vehicles successfully pass through the intersection. This final state is defined as the normal terminal state and thus terminates the simulation process. However, there are some special intermediate states, such as vehicle crash or other kinds of traffic rule violation, after which the simulation lost its practical significance. These final states are defined as the abnormal terminal state that will also terminate the simulation process. In order to further improve the expansion efficiency of Monte Carlo tree and accelerate the rollout algorithm, we create the rollout termination module as equation (10) to identify abnormal terminal state and to shorten the simulation period duration.

Simulation terminates if

$$\min Y(t) \geq l_s + l_a, \quad t = 0, 1, \ldots, T, \tag{10a}$$

$$\min D(t) \leq 0, \quad t = 0, 1, \ldots, T, \tag{10b}$$

$$\exists y_i(t) \in Y(t),\ t \in \left[n * t_c + t_g + t_y,\ (n+1) * t_c\right],\quad n = 0, 1, \ldots, \tag{10c}$$

$$\min V(t) < 0, \quad t = 0, 1, \ldots, T, \tag{10d}$$

$$\max V(t) \geq v_s, \quad t = 0, 1, \ldots, T. \tag{10e}$$

This module includes the following cases from equations (10a)–(10e): all vehicles pass the stop line, crash, running red light, reversing, and speeding. The module can avoid unnecessary simulation to reduce unnecessary expansion of the search tree to improve the efficiency of the algorithm. Figure 4 shows the influence of the rollout termination module on the structure of the search tree. It can be seen that unnecessary tree expansion has been cut after filtering, and the width and depth of the Monte Carlo tree are effectively narrowed.

3.2.4. MCTS-AVS Model. Based on the above modules, the framework of MCTS-AVS algorithm was improved over naïve MCTS algorithm (or the direct application of MCTS algorithm, denoted as n-MCTS) as shown in Figure 5.

The model works with the following steps.

(1) Start from a current state $s(t) = \{Y(t), V(t)\}$, in which $Y(t)$ is the set of all vehicles' distance from the start position, and $V(t)$ is the set of all vehicles' velocity at time t.

(2) Tree-expansion determination model determines if it is necessary to launch MCTS algorithm via equations (8) and (9). If yes, go to step 4, otherwise go to step 3.

(3) Move CAV one step ahead, and update the states of CAV and HDV accordingly. Then, go back to step 1.

(4) Determine if the maximum number of iterations has been reached. If yes, go to step 5, otherwise go to step 6.

(5) Update the states of CAV and HDV accordingly, then go back to step 1.

(6) Do *Selection*: determine the optimal action for CAV with the UCT function via equation (6). Update the states of CAV and HDV.

(7) Do *Expansion*: randomly select a move for CAV to expand the tree.

(8) Do *Simulation*: update the states of CAV and HDV, and rollout termination module determine if it is a final state via equations (10a)–(10e). If not, go to step 9. Otherwise, go to step 10.

(9) Select the next random move, and go back to step 8.

(10) Do *Backpropagation*: calculate the final benefit of X and update the node value. Then, go back to step 4.

4. Case Study

In this section, the proposed MCTS-AVS algorithm was implemented and tested on a typical arterial roadway segment with signal control. Considering that the minimum intersection spacing along an arterial corridor was usually set to be a quarter mile, the test scenario consisted of a 400-meter roadway with a signal-controlled intersection. Considering the typical congestion on the urban roadway network and the queuing process at intersection, a free flow speed of 8.33 m/s (i.e., roughly 20 mph) was used. After decomposition, CAV became the leading vehicle with a platoon of following HDVs. The platoon had six vehicles that are evenly distributed near the roadway entrance. This scenario was shown in Figure 6, and the specific parameters were listed in Table 1. Then, in MCTS-AVS algorithm, the first vehicle in the platoon was assigned as the CAV. The objective function was set to be minimization of CPI.

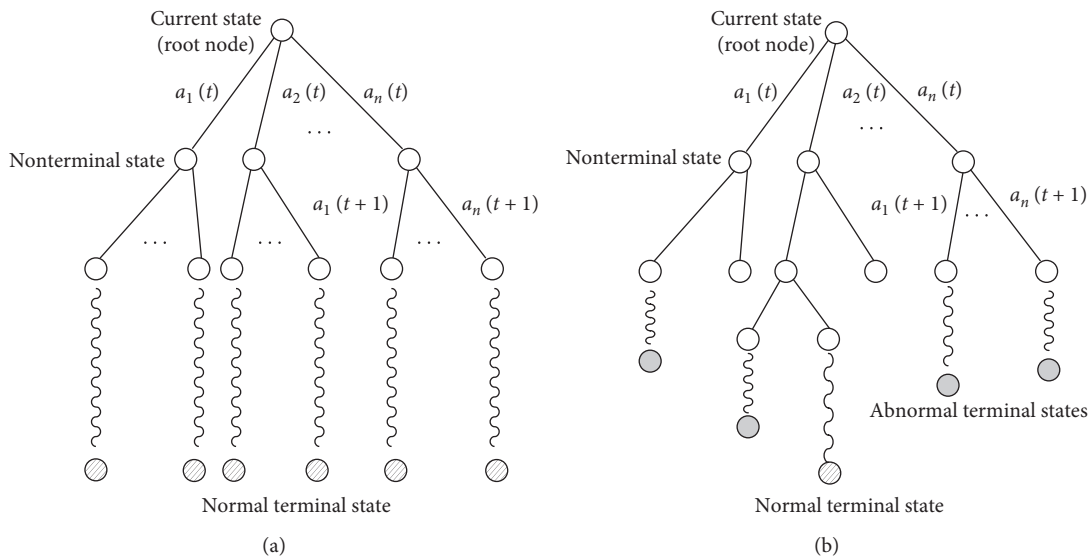


Figure 4: Comparison of tree structure: (a) without rollout termination module and (b) with rollout termination module.

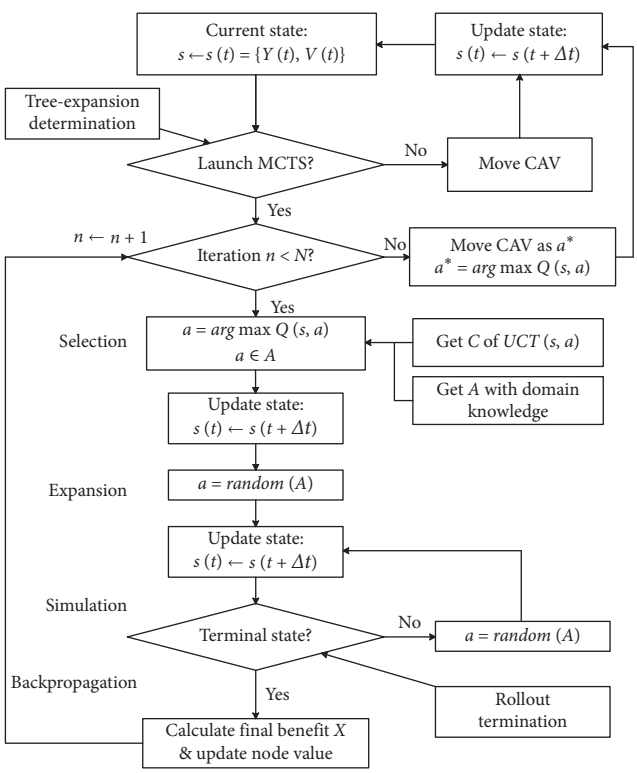


Figure 5: The proposed MCTS-AVS algorithm framework.

4.1. Algorithm Result Analysis. For comparison purpose, we defined two benchmark scenarios. The first benchmark scenario had no CAV intelligence, i.e., the CAV drove just like a typical human-driven vehicle. In other words, this first benchmark scenario was equivalent to a pure HDV scenario. The second benchmark scenario used the MCTF-MTF algorithm that was previously developed by the research team [34]. This second benchmark model, however, was developed with the objective of minimize fuel consumption and travel time of the mixed traffic flow, which makes the comparison with this newly proposed model interesting and demonstrates the safety benefits of this new MCTS-AVS algorithm.

We used the total CPI value minimization as the objective function and found the CPI value dropped from 162.63 in the benchmark scenario (without any CAV intelligence) to 38.12 with the proposed algorithm. In other words, the CPI value was reduced by 76.56%. This benefit was also greater than the previous MCTS-MTF approach, which had a CPI value of 43.36. In other words, when compared with the second benchmark model, a CPI saving of 12.08% was achieved. The capabilities of CPI were also evidenced by the time-space diagram in Figure 7.

In Figure 7, Figure 7(a) represents the benchmark scenario without any CAV intelligence, in which we can see the vehicles firstly drove at a constant high speed to the intersection, then braked and stopped at the intersection due to red light, and finally accelerated and passed intersection when the light turned green. Drastic braking of the lead vehicle caused a series of deceleration of the following HDVS, which significantly increased the crash potential of this traffic flow. On the contrary, a much smoother trajectory was found in Figure 7(b), as this proposed MCTS-AVS algorithm avoided sharp deceleration and acceleration and ensured that CPI value of mixed traffic was kept as low as possible. Figure 7(c) shows a less smooth curve of the previously developed MCTS-MTF method. However, the effect on safety improvement of the previous method was still lower than MCTS-AVS.

4.2. Algorithm Convergence Analysis. Figure 8 below shows the changes in the CPI value at different iterations. The convergence curve shows that CPI value dropped significantly to 38.39 (46.8%) when the number of iterations increased to 25. After that, the results fluctuated with the

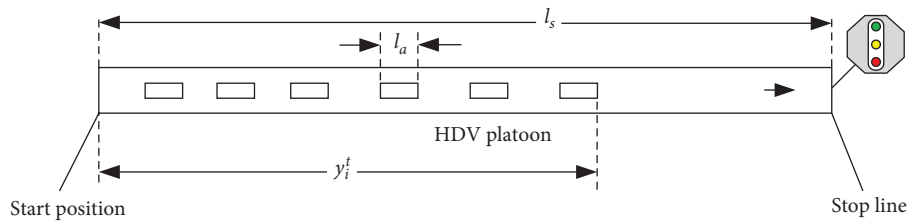


Figure 6: The component of case study benchmark.

Table 1: Environment variables and hyperparameters.

Variables and hyperparameters	Descriptions and values
l_{seg}	Roadway length, 400 meters
l_a	Average vehicle length, 5 meters
v_s	20 kph
Signal cycle	$t_g = 30s$, $t_y = 5s$, $t_r = 35s$, and $t_c = 70s$
C	Balance parameter, 0.08
N	Iteration times, 50

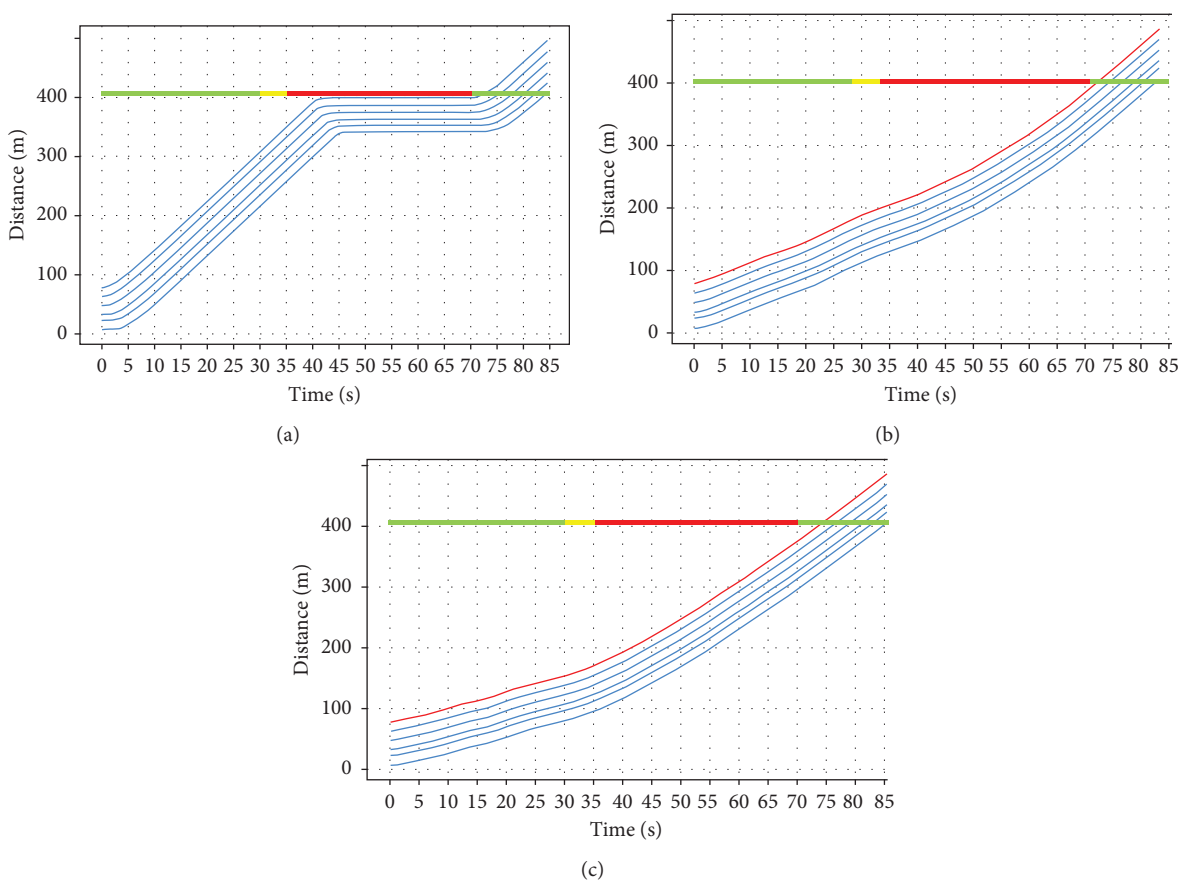


Figure 7: Time-space diagram comparison. (a) Benchmark scenario, (b) MCTS-AVS method, and (c) MCTS-MTF method.

increase of iterations. It was also observed that, after 50th iteration, the CPI value actually became very stable, the degree of fluctuation was less than 1, i.e., within 1/73 = 1.37% and can be considered as converged.

4.3. Background Traffic Sensitive Analysis. The algorithm's performance in the reducing CPI value was further tested with varying level of service (LOS, 1 ~ 6 corresponds to A ~ F), and the results were shown in Figure 9 and Table 2. It

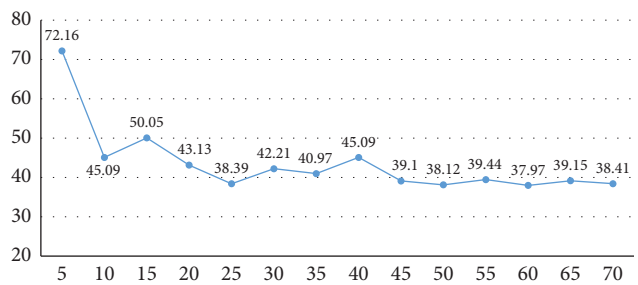


Figure 8: MCTS-AVS convergence analysis.

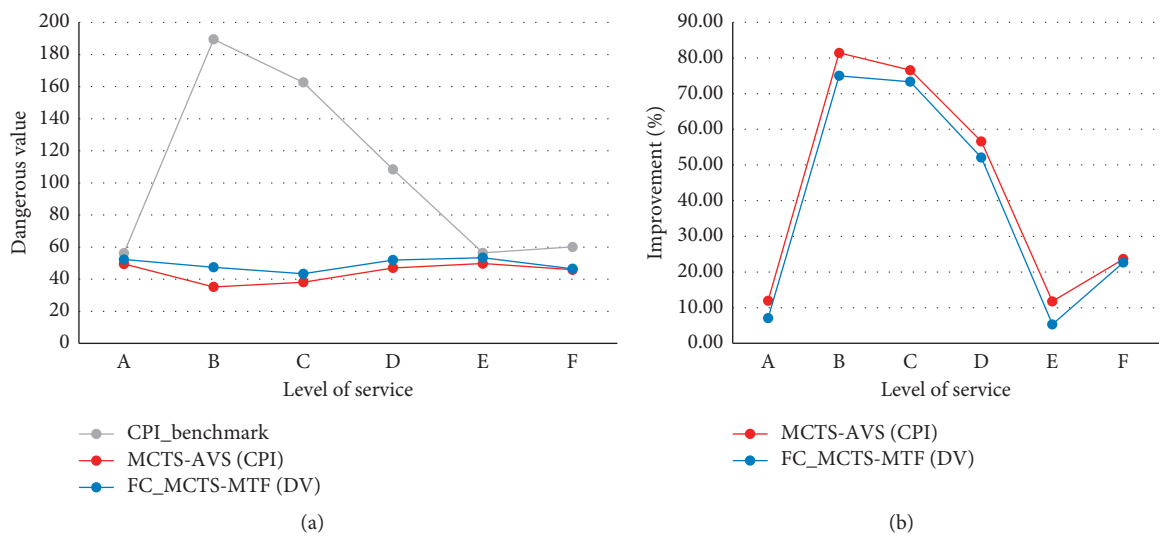


Figure 9: Algorithm performance in different LOS. (a) absolute values and (b) saving percentages.

Table 2: Algorithm performance comparison in different LOS.

LOS	Benchmark1 (no intelligence)	Proposed model: MCTS-AVS	Benchmark2: (MCTS-MTF
A	56.24	49.526 (11.94%)	52.26 (7.08%)
B	189.54	35.22 (81.42%)	47.41 (74.99%)
C	162.63	38.12 (76.56%)	43.36 (73.34%)
D	108.42	47.06 (56.59%)	51.96 (52.08%)
E	56.4	49.76 (11.77%)	53.41 (5.30%)
F	60.12	45.91 (23.64%)	46.5 (22.65%)

can be found that, for the CPI value, the maximum saving was observed at LOS B, while near minimum saving was observed at LOS A, E, and F. The guess was that when the traffic was free flowing (e.g., LOS A), not much can be done to reduce the CPI value. On the contrary, there was also a greater risk of collision during a free-flowing traffic (e.g., LOS B) decelerating process due to the change of signal light. Whereas when traffic was congested (i.e., LOS E and F), the percentage of saving was reduced significantly considering slowly moving and a low risk of collision between vehicles.

5. Conclusion and Future Research

This manuscript presents a reinforcement learning modeling approach, named Monte Carlo tree search-based autonomous vehicle safety algorithm, or MCTS-AVS, to optimize the safety of mixed traffic flow, on a one-lane roadway with signalized intersection control. Crash potential index is defined to quantitively measure the safety performance of the traffic flow. The CAV trajectory planning problem is formulated as an optimization model, and the solution

procedure is proposed. The tree-expansion determination module and rollout termination module are developed to identify and reduce the unnecessary tree expansion, so as to train the model more efficiently towards the desired direction. The case study results found that the proposed algorithm was able to reduce the CPI by 76.56%, when compared with a benchmark model without any intelligence, and 12.08% when compared with another benchmark model which the team developed earlier. These results demonstrated the satisfactory performance of the proposed algorithm in enhancing the safety of the traffic flow.

In order to expand the research scenario from one-lane traffic to a general roadway with multiple lanes, future research may be focused on the following topics. First, how to decompose this mixed traffic to satisfy the proposed algorithm or become a cornerstone of algorithm improvement is a topic worth investigation. Furthermore, with the increase of the number of lanes, there is not only car-following behavior but also lane-changing movements with greater randomness of this scenario. From the algorithm itself, how to improve the simulation efficiency and identify the unnecessary tree expansion node under the complex conditions can also be investigated.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[2] H. Qi, R. Dai, Q. Tang, and X. Hu, "Coordinated intersection signal design for mixed traffic flow of human-driven and connected and autonomous vehicles," *IEEE Access*, vol. 8, pp. 26067–26084, 2020.

[3] H. Qi, R. Dai, Q. Tang, and X. Hu, "Quasi-real time estimation of turning movement spillover events based on partial connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 120, Article ID 102824, 2020.

[4] Q. Tang, Y. Cheng, X. Hu, C. Chen, Y. Song, and R. Qin, "Evaluation methodology of leader-follower autonomous vehicle system for work zone maintenance," *Transportation Research Record: Journal of the Transportation Research Board*, 2020.

[5] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 683–711, 2021.

[6] N. Chen, M. Wang, T. Alkim, and B. Van Arem, "A robust longitudinal control strategy of platoons under model uncertainties and time delays," *Journal of Advanced Transportation*, vol. 201813 pages. In press(PT.2), Article ID 9852721, 2018.

[7] Z. Wu, Z. Gao, W. Hao, and J. Ma, "An optimal longitudinal control strategy of platoons using improved particle swarm optimization," *Journal of Advanced Transportation*, vol. 2020, Article ID 8822117, 2020.

[8] A. A. Malikopoulos, L. Beaver, and I. V. Chremos, "Optimal time trajectory and coordination for connected and automated vehicles," *Automatica*, vol. 125, no. 6, p. 109469, 2021.

[9] D. Chen, S. Ahn, M. Chitturi, and D. A. Noyce, "Towards vehicle automation: roadway capacity formulation for traffic mixed with regular and automated vehicles," *Transportation Research Part B: Methodological*, vol. 100, pp. 196–221, 2017.

[10] A. Ghiasi, X. Li, and J. Ma, "A mixed traffic speed harmonization model with connected autonomous vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 210–233, 2019.

[11] H. Jiang, J. Hu, S. An, M. Wang, and B. B. Park, "Eco approaching at an isolated signalized intersection under partially connected and automated vehicles environment," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 290–307, 2017.

[12] M. Pourmehrab, L. Elefteriadou, S. Ranka, and M. Martin-Gasulla, "Optimizing signalized intersections performance under conventional and automated vehicles traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, 2017.

[13] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Rolling horizon control framework for driver assistance systems. Part I: mathematical formulation and non-cooperative systems," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 271–289, 2014.

[14] K. Yang, S. I. Guler, and M. Menendez, "Isolated intersection control for various levels of vehicle technology: conventional, connected, and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, 2016.

[15] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: a reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 433–443, 2020.

[16] X. He, H. X. Liu, and X. Liu, "Optimal vehicle speed trajectory on a signalized arterial with consideration of queue," *Transportation Research Part C: Emerging Technologies*, vol. 61, pp. 106–120, 2015.

[17] X. Wu, X. He, G. Yu, A. Harmandayan, and Y. Wang, "Energy-optimal speed control for electric vehicles on signalized arterials," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2786–2796, 2015.

[18] X. Liang, S. I. Guler, and V. V. Gayah, "Joint optimization of signal phasing and timing and vehicle speed guidance in a connected and autonomous vehicle environment," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2673, no. 4, pp. 70–83, 2019.

[19] H. Yao, J. Cui, X. Li, Y. Wang, and S. An, "A trajectory smoothing method at signalized intersection based on individualized variable speed limits with location optimization," *Transportation Research Part D: Transport and Environment*, vol. 62, pp. 456–473, 2018.

[20] J. Ma, X. Li, F. Zhou, J. Hu, and B. B. Park, "Parsimonious shooting heuristic for trajectory design of connected automated traffic part II: computational issues and optimization," *Transportation Research Part B: Methodological*, vol. 95, no. 2341, pp. 421–441, 2017.

[21] F. Zhou, X. Li, and J. Ma, "Parsimonious shooting heuristic for trajectory design of connected automated traffic part I:

theoretical analysis with generalized time geography," *Transportation Research Part B: Methodological*, vol. 95, no. 5, pp. 394–420, 2017.

[22] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, 2012.

[23] K. Ahn, H. A. Rakha, and S. Park, "Ecodrive application: algorithmic development and preliminary testing," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2341, no. 1, pp. 1–10, 2013.

[24] Z. Cao, S. Jiang, J. Zhang, and H. Guo, "A unified framework for vehicle rerouting and traffic light control to reduce traffic congestion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1958–1973, 2017.

[25] X. Liang, S. I. Guler, and V. V. Gayah, "An equitable traffic signal control scheme at isolated signalized intersections using connected vehicle technology," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 81–97, 2020.

[26] X. Liang, S. I. Guler, and V. V. Gayah, "Traffic signal control optimization in a connected vehicle environment considering pedestrians," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 10, pp. 499–511, 2020.

[27] A. Mirheli, L. Hajibabai, and A. Hajbabaie, "Development of a signal-head-free intersection control logic in a fully connected and autonomous vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 412–425, 2018.

[28] A. Mirheli, M. Tajalli, L. Hajibabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 161–176, 2019.

[29] E. Walraven, M. T. J. Spaan, and B. Bakker, "Traffic flow optimization: a reinforcement learning approach," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 203–212, 2016.

[30] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.

[31] D. C. Gazis, R. Herman, and R. W. Rothery, "Non-linear follow-the-leader models of traffic flow," *Operations Research*, vol. 9, no. 4, pp. 545–567, 1961.

[32] C. B. Browne, E. Powley, D. Whitehouse et al., "A survey of Monte Carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[33] H. Qi and X. Hu, "Monte Carlo tree search-based intersection signal optimization model with channelized section spillover," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 281–302, 2019.

[34] Y. Cheng, X. Hu, Q. Tang, H. Qi, and H. Yang, "Monte Carlo tree search-based mixed traffic flow control algorithm for arterial intersections," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 8, pp. 167–178, 2020.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 2026895, 9 pages
https://doi.org/10.1155/2021/2026895

Research Article

An Efficient and Fast Model Reduced Kernel KNN for Human Activity Recognition

Zongying Liu, Shaoxi Li, Jiangling Hao, Jingfeng Hu, and Mingyang Pan

Dalian Maritime University, Faculty of Navigation, No. 1 Linghai Road, Dalian 116085, China

Correspondence should be addressed to Shaoxi Li; lishaoxi@dlmu.edu.cn

Received 8 April 2021; Accepted 26 May 2021; Published 3 June 2021

Academic Editor: Chunjia Han

With accumulation of data and development of artificial intelligence, human activity recognition attracts lots of attention from researchers. Many classic machine learning algorithms, such as artificial neural network, feed forward neural network, *K*-nearest neighbors, and support vector machine, achieve good performance for detecting human activity. However, these algorithms have their own limitations and their prediction accuracy still has space to improve. In this study, we focus on *K*-nearest neighbors (KNN) and solve its limitations. Firstly, kernel method is employed in model KNN, which transforms the input features to be the high-dimensional features. The proposed model KNN with kernel (K-KNN) improves the accuracy of classification. Secondly, a novel reduced kernel method is proposed and used in model K-KNN, which is named as Reduced Kernel KNN (RK-KNN). It reduces the processing time and enhances the classification performance. Moreover, this study proposes an approach of defining number of *K* neighbors, which reduces the parameter dependency problem. Based on the experimental works, the proposed RK-KNN obtains the best performance in benchmarks and human activity datasets compared with other models. It has super classification ability in human activity recognition. The accuracy of human activity data is 91.60% for HAPT and 92.67% for Smartphone, respectively. Averagely, compared with the conventional KNN, the proposed model RK-KNN increases the accuracy by 1.82% and decreases standard deviation by 0.27. The small gap of processing time between KNN and RK-KNN in all datasets is only 1.26 seconds.

1. Introduction

Since the 21st century, the development of Internet and the popularity of wearable devices brings data explosion. These massive data provide the foundation of structuring artificial intelligence (AI) algorithms. In recent decades, machine learning algorithms as backbone of AI make great progress in many domains, for example, wind speed prediction in Brazil by machine learning algorithms [1], weather time series prediction by fuzzy models [2], automatic sleep stage classification by convolution neural network [3], and the diseases classification [4, 5].

At the same time, with development of microprocessor and enabling sensors with high computational ability, the small size, and low cost, portable devices, such as smartphone, band, smart watch, or professional sensors, are used widely and record the huge data from users. Moreover, machine learning algorithms successfully solve problems in our daily life [6–8]. Nowadays, activity recognition using wearable sensors data attracts attention from researchers and businessman. Machine learning algorithms are also applied in variant intelligent devices, such as intelligent band with activity detection element that records the different activity situation for keeping health, or iWatch with fall detection function, which assists in monitoring older people activities and alarming dangerous activities. The state-of-the-art classifiers successfully achieve in human activity recognition. Artificial neural network (ANN) is a widely used model that models the relationship between input and output units. The most popular training algorithm in ANN is the backpropagation which iterative learns a set of weights for the prediction of the class labels. Paper [9] also applies optimal ANN classifier to recognize human activity based on mobile sensors data. However, the backpropagation

algorithm takes long time for seeking the suitable weights of ANN. It limits the environment of application. The other famous model is called support vector machines (SVM). Palaniappan et al. [10] combined the widely used approach SVM with a novel scheme of representing human activities for classifying human activity. Due to kernel computation, SVM also needs to face the same problem as model ANN. Besides, k-nearest neighbors (KNN) is a simple, yet effective classification recognition machine learning algorithm that is widely applied. It is applied to build a human activity recognition system and obtains a significantly outstanding performance [11].

Although KNN has less computation than other classic machine learning algorithms, it still has limitations in the processing of classification. There are three main aims in this study. The first one is to enhance the classification performance comparing with the conventional KNN. This study employs kernel method to transform the input data to be high-dimensional features. Secondly, KNN will face the heavy computation when data is large scaled. Thus, the second aim is to propose a novel reduced kernel to decrease the processing time and further increase classification accuracy. Lastly, the parameter dependency problem in KNN is a hot topic. This study proposes the way of defining the number of K neighbors that achieves the best classification performance than others. Therefore, four contributions of this study are briefly described as follows:

(i) The kernel method is applied in KNN for gaining the high-dimensional features that impact the classification performance in the positive way.

(ii) A novel reduced kernel approach is proposed. It is successfully applied to reduce the heavy computation of kernel method and increase the computing efficiency. An efficient and fast model called Reduced Kernel KNN is proposed.

(iii) A method of defining number of K neighbors is proposed, which obtains the outstanding performance in classification compared to others.

(iv) The proposed model not only has the superior ability in human activity recognition, but also achieves the better performance in benchmarks compared with other models. Thus, it has generalized ability in classification.

The paper is organized as follows. Section 2 reviews the conventional model KNN. Section 3 describes our new methods and proposed model. Section 4 explains the experiment setup and evaluation of our algorithm against baselines and proves the modifying model has influence on the classification performance. Section 5 provides conclusion.

2. Preliminary Works

KNN attracts lots of attention from researchers and project developers. Due to its easy-to-implement characteristic, it is widely applied to solve both classification and regression problems, such as financial time series prediction [12], short-term traffic flow prediction [13], recognition of diseases in paddy leaves [14], and human activity recognition [11]. Although it achieves the good performances in the different domains, it still has limitation for dealing with large-scaled data. The computation complexity will increase in KNN when the size of data in use grows. Before introducing our proposed method, let us briefly represent the working process of KNN first.

In classification aspect, KNN applies feature distance to predict the coming sample how closed to the points in the training set. It categorizes the sample with the closest feature distance into the specific category. The output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its K-nearest neighbors. Generally, K is a positive integer and small value in KNN. The working process is introduced as the following steps: firstly, we assume that $X = \{x_{1,1}, \ldots, x_{i,j}\} \in \mathfrak{R}^{N\times P}$ is the input features and $i = \{1, 2, \ldots, N\}$, $j = \{1, 2, \ldots, D\}$, where N represents the size of data and P is the number of features. At the same time, Y is a vector that represents the labels of the corresponding input features. Secondly, the value of K is supposed to define by user, which directly impacts the performance of classification. Next, it calculates the distance between the training samples and the current sample from the texting data. Generally, to avoid the matching problem between objects, the distance is computed by Euclidean Distance equation. Assume that there are two objects a and b with N-dimension. The equation of distance is shown as follows:

$$D(a,b) = \sqrt{\left(\sum_{i=1}^{N} a_i - b_i\right)^2}. \tag{1}$$

And, set an index based on all distances and then sort them in ascending order. Then, it will choose the top K rows from the sorted index. And then, keep the top K rows from all classes in the sorted index. Finally, return the class label based on the most frequent class of processed index. Pseudocode for KNN is shown in Algorithm 1.

Based on previous research studies on the conventional KNN and variant KNN, many research works played a vital role in classification. For example, a multi-label lazy learning approach based on KNN, which mainly applied maximum a posteriori (MAP) principle to gain statistical information from training data, determines the label set for the unseen instance [15]. Moreover, the ensemble algorithm based on KNN also attracted attention. Xiao proposed an ensemble learning algorithm that applied support vector machine and KNN for traffic incident detection [16]. On the other hand, researches paid more attention to the problem of definition of parameter K in variant KNN, which directly impacted the performance in classification. Zhang et al. [17] proposed a S-KNN algorithm to identify an optimal K value. Sharma [18] applied DBSCAN to set parameter according to information of the data as it got accumulated in a cluster structure.

Therefore, according to the characteristic of KNN and variant KNN, they have two main drawbacks existing in the process of classification. The parameter of KNN and variant

Require: input data matrix, X; the target value, Y; the parameter of KNN, K; the number of training data, L.
Ensure: the forecasting output ($\widehat{Y}$).
Data Separation:
(1) The training features: $X_{\text{tr}} = [x_1, x_2, \ldots, x_L]$;
(2) The training labels: $Y_{\text{tr}} = [y_1, y_2, \ldots, y_L]$;
(3) The testing features: $X_{\text{tx}} = [x_{L+1}, x_{L+2}, \ldots, x_N]$;
(4) The texting labels: $Y_{\text{tx}} = [y_{L+1}, y_{L+2}, \ldots, y_N]$;
Loop:
(5) **for** $t \in \{1, \ldots, N-L\}$ **do**
(6) **for** $i \in \{1, \ldots, L\}$ **do**
(7) Calculate the distance D_i between $X_{\text{tr}}[i]$ and X_{tx} by equation (1);
(8) **end for**
(9) Sort the distance D from smallest to largest value (in ascending order);
(10) Pick the top K vectors from the sorted collection as an index;
(11) Set the forecasting the class label $\widehat{Y}_t$ based on the most frequent class of processed index.
(12) **end for**
return $\widehat{Y}$

Algorithm 1: The working process of KNN.

KNN impacts the performance of classification. The computation complexity will grow sharply when the size of data increases. Hence, the main aims of this study are to build an efficient and fast classification model based on KNN for solving these two problems and enhance the classification performance, especially for human activity. The following will explain the proposed methods with improvement of the classification performance and reduction of computation complexity.

3. Methodology

For the large-scaled data, the time consumption of KNN is largely researched in the machine learning. Although KNN has efficient classification, the heavy computation usually is a barrier for applying in the real-world project when the training data become large. In this section, an efficient and fast model is introduced based on the baseline model KNN for improving the classification performance in some extent and increasing the training efficiency. Firstly, data processing is described. Secondly, the model KNN with kernel method is introduced, where kernel method is applied in KNN. It employs the characteristic of kernel method for expanding the dimension of features and enhances the performance of classification in the baseline model KNN. Next, reduced kernel method is introduced. It mainly focuses on reducing the kernel computation. Finally, the detail of our proposed method is shown.

3.1. Data Processing. In this section, we assume that the dataset (X) with N samples is represented as follows:

$$X = x_{i,j} = \begin{bmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,D} & Y_1 \\ x_{2,1} & x_{2,2} & \ldots & x_{2,D} & Y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \ldots & x_{N,D} & Y_N \end{bmatrix}, \tag{2}$$

where $x_{i,j}$ is the element feature in the i-th row and j-th column from data matrix X and D is represented by the number of features. The last column of X is the corresponding i label (Y_i).

Then, the data (X) is separated by the training data (TR) and testing data (TX). Their corresponding features and labels are X_{tr}, Y_{tr} and X_{tx}, Y_{tx}, respectively. Here, we set the number of training data as L. The matrix data (X) can be processed as follows:

$$X_{\text{tr}} = \begin{bmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,D} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{L,1}, & x_{L,2} & \ldots & x_{L,D} \end{bmatrix}, \tag{3}$$

$$X_{\text{tx}} = \begin{bmatrix} x_{L+1,1} & x_{L+1,2} & \ldots & x_{L+1,D} \\ x_{L+2,1} & x_{L+2,2} & \ldots & x_{L+2,D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N,1}, & x_{N,2} & \ldots & x_{N,D} \end{bmatrix}, \tag{4}$$

where their corresponding labels are $Y_{\text{tr}} = [y_1, y_2, \ldots, y_L]$ and $Y_{\text{tx}} = [y_{L+1}, y_{L+2}, \ldots, y_N]$.

3.2. KNN with Kernel Method. Kernel method is applied in the different algorithms, such as extreme learning machine [19], support vector machine [20], online learning algorithms [21, 22], and multilayer algorithms [23, 24], which obtain the good performance in regression and classification. The kernel method transforms the low-dimensional features to the high-dimensional features. In this study, we employ Gaussian Kernel method ($G(\ldots)$) to process the features. Its mathematical equation is shown in the following equation:

$$G(A, B, \sigma) = e^{-\left(\left(A-\mu_A\right)^2 + \left(B-\mu_B\right)^2 / 2\sigma^2\right)}, \tag{5}$$

where μ_A and μ_B are the mean of input data A and B, respectively, and σ represents the kernel parameter that is defined by user.

This study employs Gaussian kernel method to provide high-dimensional features for the classification datasets. It is used to combine KNN algorithm in order to improve its classification ability. This algorithm is proposed and named as k-nearest neighbor with kernel (K-KNN). The main working process of K-KNN is separated by the four parts. Firstly, the features of data, including X_{tr} and X_{tx}, are transformed by Gaussian kernel method. The kernel matrix of training features (KX_{tr}) and testing features (KX_{tx}) is computed by equations (6) and (7), respectively:

$$KX_{\text{tr}} = G(X_{\text{tr}}, X_{\text{tr}}, \sigma), \tag{6}$$

$$KX_{\text{tx}} = G(X_{\text{tr}}, X_{\text{tx}}, \sigma). \tag{7}$$

Secondly, the distances between all testing features and training feature matrix are calculated by equation (1). Next, sort the distance for each class in ascending order and then pick the top K elements from the sorted collection index. Finally, the prediction class will be voted based on the most frequent class of processed index. The main process of K-KNN is shown in Algorithm 2.

However, the Gaussian kernel method brings the heavy computation with the increasing scale of data, which directly leads to the decrease of efficiency in model K-KNN. To solve this limitation and enhance the efficiency of computation in the process of classification, the reduced kernel method is proposed in this study. The following section will introduce the details.

3.3. Reduced KNN with Kernel Method. This section proposes a new model named Reduced KNN with kernel method (RK-KNN). It applies the reduced kernel method that replaces the intrinsic kernel method in the model K-KNN.

The main idea of reduced kernel method is to apply a certain percentage of training data from each class to calculate the kernel matrix, which directly reduces the computation complexity of kernel method. Generally, variant models with kernel method have the good performance in regression and classification. For instance, extreme learning machine with kernel (KELM) [25, 26] even obtains better performance in regression and classification than the conventional extreme learning machine. The classic model support vector machine (SVM) applies kernel method to connect the input layer with the hidden layer, which provides the high-dimensional support vectors to transform the original input features. The kernel method of the model SVM also plays a vital role in enhancing the classification performance. These algorithms use the entire input to compute the kernel matrix. This study claims that the kernel matrix is computed by the randomly selected samples from the training observations. This type of features representation is the same or even better than that of the conventional kernel matrix. The mathematical equation of reduced kernel is represented as follows:

$$\overline{KM} = G(A_P, B, \sigma), \tag{8}$$

where $\overline{KM}$ is the reduced kernel matrix, P represents the selected percentage for computing kernel matrix, and A_P stands for the selected P samples from input data A.

Then, the reduced kernel matrix of training and testing data can be calculated by equations (9) and (10), respectively:

$$\overline{KX_{\text{tr}}} = G(X_{\text{tr}\,P}, X_{\text{tr}}, \sigma), \tag{9}$$

$$\overline{KX_{\text{tx}}} = G(X_{\text{tr}\,P}, X_{\text{tx}}, \sigma), \tag{10}$$

where $X_{\text{tr}\,P}$ represents the P percentage samples from each class in the training observations X_{tr}.

The kernel matrix of K-KNN can be replaced by the reduced kernel matrix, which has less computation of kernel matrix than full kernel matrix in model K-KNN. Reduced kernel method not only keeps the high-dimensional features from kernel method, but also reduces the computing process of kernel matrix by selecting certain percentage of training samples. Therefore, the brief pseudocode of RK-KNN is shown in Algorithm 3.

4. Experimental Works

To prove that the kernel method and reduced method play a vital role in the enhancement of classification performance and decrease of processing time in model KNN, this section applies ten benchmarks dataset (binary and multiclass) and two real-world human activity datasets. Moreover, the parameter dependency exists in the variant models of KNN. The last part of this section indicates how to define the range of K parameter in our proposed model.

4.1. Data Description and Parameter Setting. Run ten benchmarks and two human activity datasets in the experiment, which include binary and multiclass.

For the real-world data, this study uses two human activity datasets in the following experiments. The first data is Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set (HAPT) [27]. It captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50 Hz using the embedded accelerator and gyroscope of the device. Figure 1 shows the condition of activities, including stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand.

The other human activity data is Smartphone Dataset for Human Activity Recognition in Ambient Assisted Living Data (Smartphone) [28]. Figure 2 shows the different human activities, including standing, sitting, laying, walking, walking upstairs, and walking downstairs.

The obtained datasets are randomly partitioned into two sets, where 70% of the volunteers are selected for generating the training data and the remaining volunteers belong to the test data.

Besides, all benchmark datasets can be found in the part of classification from UCI Machine Learning Repository. The number of features and classes in benchmark and real-world datasets is shown in Table 1.

Require: input data matrix, X; the target value, Y; the parameter of K-KNN, K; the number of training data, L; the kernel parameter, σ.
Ensure: the forecasting output ($\widehat{Y}$).
Data Separation:
(1) Data matrix (X) is separated as X_{tr} and X_{tx} by equation (3) and (4), separately;
Kernel Computation Part:
(2) Calculate the kernel matrix of training features (KX_{tr}) by equation (6);
(3) Calculate the kernel matrix of testing features (KX_{tx}) by equation (7);
Loop:
(4) **for** $t \in \{1, \ldots, N - L\}$ **do**
(5) **for** $i \in \{1, \ldots, L\}$ **do**
(6) Calculate the distance D_i between $KX_{\text{tr}}[i]$ and KX_{tx} by equation (1);
(7) **end for**
(8) Sort the distance D from smallest to largest value (in ascending order);
(9) Pick the top K vectors from the sorted collection as an index;
(10) Set the forecasting the class label $\widehat{Y}_t$ based on the most frequent class of processed index.
(11) **end for**
return $\widehat{Y}$

Algorithm 2: The working process of K-KNN.

Require:
Training data matrix X_{tr};
Testing data matrix X_{tx};
Number of training data L;
Parameter of RK-KNN K;
Kernel parameter σ;
The percentage of selected data for reducing kernel matrix P;
Ensure:
Prediction class $\widehat{Y}$.
Reduced Kernel Computation Part:
(1) Select P percentage samples for each class from the training data as $X_{\text{tr}\,P}$;
(2) Calculate the reduced kernel matrix for training features by equation (9);
(3) Calculate the reduced kernel matrix for testing features by equation (10);
Loop:
(4) **for** $t \in \{1, \ldots, N - L\}$ **do**
(5) **for** $i \in \{1, \ldots, L\}$ **do**
(6) Calculate the distance D_i between $\overline{KX_{\text{tr}}}[i]$ and $\overline{KX_{\text{tx}}}$ by equation (1);
(7) **end for**
(8) Sort the distance D in the ascending order;
(9) Pick the top K vectors from the sorted collection as an index;
(10) Set the forecasting the class label $\widehat{Y}_t$ based on the most frequent class of processed index.
(11) **end for**
return $\widehat{Y}$

Algorithm 3: The working process of RK-KNN.

Besides, the parameter setting directly affects the performance of classification. Fairly comparison among models is necessary. Firstly, KNN and all variant KNN models need to define the number of neighbors (K). The following experiments set the number of neighbors (K) as the class size for all compared models. Secondly, due to the kernel method that is used in model K-KNN and RK-KNN, the kernel parameter is set as the same number. Lastly, to reduce kernel computation, the selected percentage is defined as 10% for benchmarks and 30% for real-world data in model RK-KNN.

4.2. Experimental Results and Discussion. Based on the parameters setting, three models are compared, including baseline model KNN, KNN with kernel method, and proposed model RK-KNN. Due to random selection for training and testing samples from all datasets, all experiments are run ten times in order to keep the generalization of classification performance. Moreover, to compare training time fairly, we run all experiments by python3.6 version under Windows 10 system with 16 GB memory and Intel 8th Generation i7 processor.

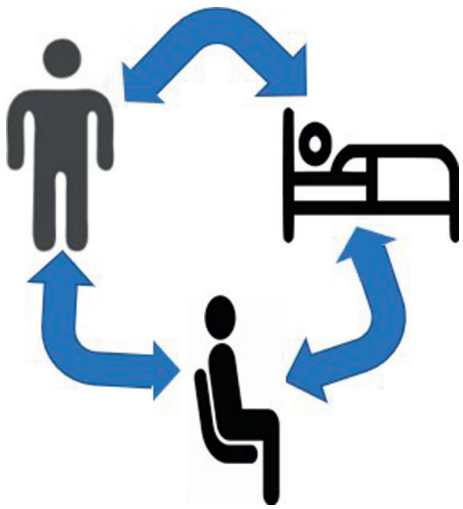


Figure 1: The type of activities in HAPT data.

Figure 2: The type of classes for human activity in Smartphone data.

Table 1: Details of datasets.

Benchmarks	Observations	Features	Classes
Iris	150	4	3
Heart	270	13	2
Breast	263	9	2
Diabetes	768	8	2
Seed	210	7	3
Thyroid	215	5	2
Banana	5300	2	2
Image	2086	18	2
Ringnorm	7400	20	2
Splice	2991	60	2
Human activity data	Observations	Features	Classes
HAPT	10926	561	6
Smartphone	5744	561	6

Table 2 shows the classification performances in benchmarks and real-world datasets. The accuracy is used to determine the ability level of classification. Standard deviation (SD) shows the difference among ten times prediction results. The processing time is shown in the last column (Time) of each model, which is recorded in seconds. Based on the comparison between the baseline model KNN and other two models, it demonstrates that the kernel method and the reduced kernel method play a vital role in aspect of enhancement of classification and processing efficiency. Firstly, comparing the performance of model KNN with that of K-KNN, model K-KNN has better performance in all datasets than that of KNN. Averagely, the accuracy of model K-KNN increases by 0.06% compared with the baseline model KNN. The maximum increase (1.17%) appears in Smartphone dataset. At the same time, the performance of K-KNN in HAPT data keeps the constant with that of model KNN. In aspect of standard deviation, the real-world datasets in the model K-KNN have lower values than those in the model KNN. There are similar differences in SD for benchmarks. However, the processing time in model K-KNN is almost ten times that of model KNN.

To keep the advantage of kernel method for enhancing classification ability and overcome the limitation of heavy kernel computation, we propose an efficient and fast model RK-KNN. The second experiment focuses on the role that

Table 2: The comparison results among three models.

Data	KNN			K-KNN			RK-KNN		
	Accuracy (%)	SD	Time (s)	Accuracy (%)	SD	Time (s)	Accuracy (%)	SD	Time (s)
Iris	92.66	2.56	0.03	93.12	4.05	0.27	**95.20**	2.73	0.05
Heart	88.51	3.13	0.21	88.57	3.49	1.61	**91.66**	2.56	0.27
Breast	81.49	4.34	0.19	82.04	4.36	1.49	**82.77**	4.37	0.49
Diabetes	83.64	2.34	2.62	83.85	2.46	44.19	**84.32**	2.23	11.67
Seed	95.21	2.39	0.09	95.53	3.02	0.73	**97.36**	1.37	0.14
Thyroid	97.41	2.20	0.08	97.48	2.09	0.79	**97.67**	1.98	0.31
Banana	93.15	0.67	31.67	94.14	0.63	177.98	**94.67**	0.61	21.49
Image	92.09	1.25	407.32	92.11	2.25	2396.5	**99.04**	0.97	422.5
Ringnorm	92.14	1.15	411.79	92.23	0.61	2221.83	**92.83**	0.98	521.13
Splice	92.95	0.77	413.50	93.31	0.66	2255.67	**93.51**	0.65	350.21
HAPT	91.02	0.67	500.43	91.02	0.58	4334.26	**91.60**	0.32	495.78
Smartphone	91.21	1.05	170.24	92.38	0.38	2325.65	**92.67**	0.49	129.20
Average	90.96	1.88	161.51	91.32	2.05	1146.75	92.78	1.61	162.77

Table 3: The data of Friedman test.

Data	KNN	K-KNN	RK-KNN
Iris	92.66	93.12	95.20
Heart	88.51	88.57	91.66
Breast	81.49	82.04	82.77
Diabetes	83.64	83.85	84.32
Seed	95.21	95.53	97.36
Thyroid	97.41	97.48	97.67
Banana	93.15	94.14	94.67
Image	92.09	92.11	99.04
Ringnorm	92.14	92.23	92.83
Splice	92.95	93.31	93.51
HAPT	91.02	91.02	91.60
Smartphone	91.21	92.38	92.67

reducing kernel method plays in model K-KNN. The performances are compared between models K-KNN and RK-KNN. Averagely, the accuracy of model RK-KNN increases by 1.46% compared with model K-KNN. Moreover, the model RK-KNN has the best performance in all datasets. The maximum increase rate appears in Heart data, which increases 3.09% compared with model K-KNN. The minimum increase (0.20%) exists in Splice. For the real-world datasets, model RK-KNN not only develops the classification accuracy, which increased by 0.58% for HAPT and 0.29% for Smartphone, but also reduces at least ten times processing time than model K-KNN. In aspect of standard deviation, averagely, model RK-KNN has the lowest standard deviation among these three compared models. It demonstrates that model RK-KNN has more stable classification ability than others. On the other hand, the average processing time of model RK-ELM in all datasets is close to that of model KNN. Especially for the real-world datasets, the processing time of model RK-KNN is less than that of model KNN. Therefore, the proposed model RK-KNN not only enhances the classification ability in the benchmarks and human activity datasets, but also reduces the kernel computation.

4.3. Statistical Analysis. In this section, it applies Friedman test to check whether the different modifying models impact the performances from all benchmarks and real-world datasets. The Friedman testing data is a group of twelve different datasets with their classification accuracy performances on different data as a result of a change in the different modifying model. The testing data are shown in Table 3. The H_o is that the classification accuracy will be the same regardless of the modifying model. We run Friedman test in IBM SPSS Statistics 21. Based on the testing result, in the 95% significant level, the p value of Friedman test is 0. Because p value is less than 0.05, we can reject H_o. Therefore, there is sufficient evidence that indicates the modifying model significantly altered the classification performance.

4.4. Impact of K Neighbors. K value in KNN and its variant models affects the classification performance. K value equal to one means that it will take one nearest neighbor and classify the query point based on its label. If value of K is extremely large, the model will underfit. Therefore, K value is the key element that directly impacts the classification result. To further analyze the rule of K neighbors for impact of classification performance and prove our statement regarding the definition of K neighbors in our proposed model, this section observes the different performances in a certain range of K neighbors of model RK-KNN.

Table 4: The distribution of K neighbors in benchmarks and real-world data.

Benchmarks	K number of neighbors						
	2	3	4	5	6	7	8
Iris	94.70	**95.20***	91.73	88.53	89.47	85.26	89.95
Heart	**91.66***	88.13	88.89	88.93	89.43	89.46	88.18
Breast	**82.77***	81.25	80.62	80.13	79.38	78.02	78.14
Diabetes	84.32*	84.28	83.45	84.58	84.39	84.46	**84.67**
Seed	96.03	**97.36***	94.01	92.66	93.78	94.71	94.02
Thyroid	**97.67***	97.09	96.71	96.09	95.92	93.08	94.42
Banana	**94.67***	94.08	93.35	93.08	93.91	93.28	93.16
Image	**99.04***	98.56	98.21	98.24	97.80	97.52	97.87
Ringnorm	**92.83***	76.87	74.81	74.74	74.21	73.82	73.56
Splice	93.51*	91.91	93.39	93.36	92.68	93.00	**95.46**
Human activity data	3	4	5	6	7	8	9
HAPT	85.93	88.32	90.51	**91.60***	90.32	86.62	84.34
Smartphone	88.60	89.23	91.28	**92.67***	87.83	87.98	85.39

Benchmarks provide the range from two to eight for K value of neighbors. The real-world dataset are in the range between three and nine for K value of neighbors. Table 4 shows the performances of the proposed model RK-KNN in the different ranges for benchmarks and real-world data. It demonstrates that the rule of K neighbor selection and the bold number is the best performance for each data in the different range of K neighbors. The result with star indicates that the accuracy is computed by the proposed model that set the number neighbors as the class size. Based on the performance in the different number of K neighbors in model RK-KNN, the majority of datasets with the class size as neighbors appear the best performance. There are two binary class datasets, including Diabetes and Splice existing the best performance in the situation of setting number of neighbors as eight. However, the second highest accuracy is from the proposed number of neighbors setting. Besides, there is small difference between the best and second accuracy, including 0.35% for Diabetes and 1.95%, respectively. Therefore, our proposed method of setting class size as neighbors in the model RK-KNN obtains the best performance in classification for the benchmarks and real-world datasets. It can be used for defining the number of neighbors in our proposed model, which directly solves the problem of parameter dependency problem.

5. Conclusions

In this paper, an efficient and novel model named as Reduced Kernel K-Nearest Neighbors is proposed. This study mainly proposes three approaches to modify model KNN and enhance the classification ability. Firstly, the kernel method is applied to transform the input data to be high-dimensional features, which directly influences the classification performance in the positive way. It is combined with model KNN and named as K-KNN. Comparing the performance of KNN with that of K-KNN, K-KNN obtained much better performance in all datasets. Secondly, to reduce the heavy computation of kernel method, a novel approach is proposed and applied in K-KNN. It is short for RK-ELM. The main objectives of proposed model RK-KNN are to increase the efficiency and classification accuracy. The last approach is method for selection correct K parameter. Our approach is easy to seek the suitable K parameter in the proposed model. Based on the experimental works, our proposed model obtains the best performance in benchmarks and human activity data. Not only does model RK-KNN have more stable classification ability than others, but its average processing time is close to that of the conventional KNN. It reduces approximately 10 times processing time compared to the model K-KNN. Moreover, the distribution of K neighbors also proved that the proposed approach is easier to set the number of neighbors in model RK-KNN, which also obtains the highest accuracy in classification. Therefore, our proposed model has super ability in human activity recognition and plays a significant role in solving the general classification ability. In the future, we will focus on the reduced percentage in our proposed reduced method, which in some extent impacts the classification performance.

Data Availability

The benchmarks and real-world data used to support the findings of this study have been deposited in the UCI Machine Learning Repository.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors have participated in conception and design, or analysis and interpretation of the data, drafted the article and revised it critically for important intellectual content, and approved the final version.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant

3132019400 and Individual Research Fund under Grant 02500119.

References

[1] K. Ali, L. Machado, and R. O. Nunes, "Time-series prediction of wind speed using machine learning algorithms: a case study osorio wind farm, Brazil," *Applied Energy*, vol. 224, pp. 550–566, 2018.

[2] E. Soares, P. Costa Jr, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Applied Soft Computing*, vol. 64, pp. 445–453, 2018.

[3] H. Phan, F. Andreotti, N. Cooray, O. Y. Chén, and M. De Vos, "Joint classification and prediction cnn framework for automatic sleep stage classification," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1285–1296, 2018.

[4] S. K. Lakshmanaprabu, K. S. Sachi Nandan Mohanty, N. Arunkumar, and G. Ramirez, "Optimal deep learning model for classification of lung cancer on ct images," *Future Generation Computer Systems*, vol. 92, pp. 374–382, 2019.

[5] S. Khan, N. Islam, Z. Jan, I. Ud Din, and J. Rodrigues, "A novel deep learning based framework for the detection and classification of breast cancer using transfer learning," *Pattern Recognition Letters*, vol. 125, pp. 1–6, 2019.

[6] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 Acm Sigsac Conference on Computer and Communications Security*, pp. 1528–1540, Vienna, Austria, October 2016.

[7] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019.

[8] B. M. Henrique, V. A. Sobreiro, and H. Kimura, "Literature review: machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[9] H. Kishor, V. Rajiv, and M. Vilas, "Pca based optimal ann classifiers for human activity recognition using mobile sensors data," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems*, vol. 1, pp. 429–436, Springer, Berlin, Germany, May 2016.

[10] A. Palaniappan, R. Bhargavi, and V. Vaidehi, "Abnormal human activity recognition using svm based approach," in *Proceedings of 2012 international conference on recent trends in information technology*, IEEE, Chennai, Tamil Nadu, India, April 2012.

[11] S. Sani, N. Wiratunga, S. Massie, and K. Cooper, "Knn sampling for personalised human activity recognition," *Case-Based Reasoning Research and Development*, vol. 99, pp. 330–344, 2017.

[12] L. Tang, H. Pan, and Y. Yao, "Pank-a financial time series prediction model integrating principal component analysis, affinity propagation clustering and nested k-nearest neighbor regression," *Journal of Interdisciplinary Mathematics*, vol. 21, no. 3, pp. 717–728, 2018.

[13] L. Zhao, W. Du, D.-mei Yan, C. Gan, and J.-hua Guo, "Short-term traffic flow forecasting based on combination of k-nearest neighbor and support vector regression," *Journal of Highway and Transportation Research and Development (English Edition)*, vol. 12, no. 1, pp. 89–96, 2018.

[14] M. Suresha, K. N. Shreekanth, and B. V. Thirumalesh, "Recognition of diseases in paddy leaves using knn classifier," in *Proceedings of 2017 2nd International Conference for Convergence in Technology (I2CT)*, pp. 663–666, IEEE, Mumbai, India, April 2017.

[15] M.-L. Zhang, Z. Ml-knn, and Z.-H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[16] J. Xiao, "Svm and knn ensemble learning for traffic incident detection," *Physica A: Statistical Mechanics and Its Applications*, vol. 517, pp. 29–35, 2019.

[17] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, "A novel knn algorithm with data-driven K parameter computation," *Pattern Recognition Letters*, vol. 109, pp. 44–54, 2018.

[18] A. Sharma and A. Sharma, "Knn-dbscan: using k-nearest neighbor information for parameter-free density based clustering," in *Proceedings of 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pp. 787–792, IEEE, Kannur, Kerala, India, June 2017.

[19] G.-B. Huang, Q.-Yu Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, vol. 2, pp. 985–990, Budapest, Hungary, July 2004.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[21] Z. Liu, C. K. Loo, K. Pasupa, and M. Seera, "Meta-cognitive recurrent kernel online sequential extreme learning machine with kernel adaptive filter for concept drift handling," *Engineering Applications of Artificial Intelligence*, vol. 88, Article ID 103327, 2020.

[22] Simone Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, "Online sequential extreme learning machine with kernels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2214–2220, 2014.

[23] Z. Liu, K. Chu, and K. Pasupa, "A novel error-output recurrent two-layer extreme learning machine for multi-step time series prediction," *Sustainable Cities and Society*, vol. 10, Article ID 102613, 2020.

[24] C. M. Wong, C. Man Vong, K. Pak Wong, and J. Cao, "Kernel-based multilayer extreme learning machines for representation learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 3, pp. 757–762, 2016.

[25] Z. Liu and K. Chu, N. Masuyama and K. Pasupa, "Recurrent kernel extreme reservoir machine for time series prediction," *IEEE Access*, vol. 6, pp. 19583–19596, 2018.

[26] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

[27] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[28] D. Dua and C. Graff, "UCI machine learning repository," 2017.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 9970851, 12 pages
https://doi.org/10.1155/2021/9970851

Research Article

Invalid Signatures Searching Bitwise Divisions-Based Algorithm for Vehicular Ad-Hoc Networks

Xin Ye,[1] Gencheng Xu,[2] Xueli Cheng,[2] Jin Zhou,[1] and Zhiguang Qin[1]

[1]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 610054, China*
[2]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

Correspondence should be addressed to Gencheng Xu; xugencheng@std.uestc.edu.cn

Received 4 March 2021; Accepted 12 May 2021; Published 24 May 2021

Academic Editor: Chi-Hua Chen

Vehicular ad-hoc networks (VANETs) are the crucial part of intelligent transportation systems (ITS), which are brought to enhance the security, efficiency, and comfort of transportation. VANETs have aroused extensive attention in the world recently. One of the challenges in practice is real time and low delay, which strongly requires VANETs to be efficient. Existing schemes have properly solved the problem which is how to aggregate the signatures and verify the aggregated signature. However, few solutions are proposed to pinpoint all invalid signatures if existing. The algorithms that can find all invalid signatures are not efficient enough. Following consideration of the above deficiencies of existing approaches, this paper proposes a factorial bitwise divisions (FBD) algorithm and its optimized version and early-stopping factorial bitwise divisions (EFBD) algorithm. Both algorithms are parallel-friendly. Compared with the binary-based batch verification algorithm, the experimental results demonstrate that the proposed algorithms achieve better performance in both theory and practice at low invalid signatures' rate. Especially, in the parallel condition, when the number of invalid signatures is 1, the proposed algorithms cost only one aggregation-verification delay, while the comparison is more than $\log_2 n$ times.

1. Introduction

Vehicular ad-hoc networks (VANETs) were created by applying the principles of the mobile ad-hoc networks (MANETs), which is a spontaneous creation of wireless networks for data exchange in the vehicle domain. In the early 21st century, VANETs were considered direct implementation of the MANETs, and since then, because of the particularity of VANETs, they have developed into an independent field of study. By 2015, the focus was still on spontaneous networking, not to mention roadside units (RSUs) or cellular networks. The vehicle that joins the network becomes a wireless node or a wireless route. It allows vehicles spaced 100 to 300 meters from each other to form a decentralized and dynamic network automatically. When one vehicle is out of range and off the network, other vehicles can join and link to each other, creating new mobile networks [1–3]. Nowadays, VANETs have become an important part of the Internet of things (IoT) and they have a broad application prospect.

Although it seems that VANETs own a lot of merits, there are still many problems to overcome to achieve a wide application. One of the obstacles is that VANETs require strong timeliness. Existing single verification methods can verify a signature one by one. However, in the area of heavy traffic, there are too many vehicles to be fully verified, which may lead to potential hazards. Batch verification and aggregate verification emerged at the right moment. In the batch verification mode, signatures are aggregated simultaneously at first; then, the aggregated signatures need to be verified. Compared with the validation process, the aggregation process takes very little time that it can be ignored. In this approach, if the aggregate signature can be verified, then all individual signatures are considered legal. However, all

the individual signatures will be rejected when there exists one or more illegal signature, which will cause a great waste of time and cannot take full advantage of batch verification [4–6].

In recent years, some researchers try to propose a few solutions to the aforementioned problem. Some just discard all signatures or verify them individually. Huang et al. [7, 8] use binary search to solve the problem. Indeed, it is easy to understand. Huang et al. [9] proposed a matrix-detection algorithm to reduce the escape probability of bad signatures, but it only applies to a batch of signatures which has less than four or odd number of bad signatures. However, the method still fails to achieve high efficiency. Ferng et al. [10] proposed a dynamic batch verification scheme. However, it cannot fully address the problem and has a relatively low efficiency. In addition, in the binary search algorithm, the number of layers is $\log_2 n$ which is not parallel-friendly. Therefore, binary search cannot have a relatively good performance in practice.

To solve the problem, this paper proposes a factorial bitwise divisions' algorithm (FBD) and an early-stopping factorial bitwise divisions' (EFBD) algorithm. According to the index of each signature, they can be divided into a few groups; the details of the grouping strategy will be covered in Section 4. Then, each group will be verified aggregately. If the aggregate validation passes, all signatures in this group are supposed to be legitimate. Therefore, in the later aggregate verification process, we can exclude all signatures in this group. In Section 4, we show that the size of each group is half the size of the input. Thus, half of the signatures are verified at least. Eventually, we will get a set including all illegitimate signatures and $n(n \geq 0)$ good signatures. The aforementioned algorithm is recursively executed until the input and output are completely same. Later, two algorithms execute differently. FBD will reorder each group, make them as the input, and recursively execute the algorithm, whereas EFBD will singly verify all signatures in the final output set when the size of the set is less than $2 \log_2 n$. Our main contributions in this paper are as follows:

(i) Propose FBD and EFBD algorithms which are recursive and parallel-friendly for aggregate verification in VANETs and adopt a classical bitwise method

(ii) Through simulation experiments, the proposed algorithms are effective in a real application scenario

The rest of this paper is organized as follows. In Section 2, we will introduce the related work, mainly about how the binary search algorithm works in batch verification. In Section 3, we describe some related concepts in VANETs and the problem definition of our work. Then, we present our proposed algorithms in Section 4 in detail. Time complexity analysis is described in Section 5. Experiments are performed and analyzed in Section 6. Finally, we will draw a conclusion of this paper in Section 7.

2. Related Works

In this section, we will cover several famous grouping strategies, especially the binary search algorithm.

Heidari et al. [11] propose a novel population-based, nature-inspired optimization paradigm, which is called Harris Hawks Optimizer (HHO). Yang et al. [12] propose a general-purpose population-based optimization technique called Hunger Games Search (HGS). It has a simple structure, special stability features, and very competitive performance to realize the solutions of both constrained and unconstrained problems more effectively. Li et al. [13] propose a new stochastic optimizer called slime mould algorithm (SMA) which is based on the oscillation mode of slime mould in nature.

To detect all invalid requests, Huang et al. [8] proposed a binary divisions' detection (BDD) algorithm, which is based on the "divide-and-conquer" approach. It is shown in Algorithm 1. When BDD is applied in the signature verification process, it can easily and efficiently find all invalid signatures, and it is easy to understand. A full BDD tree is shown in Figure 1, where the dark red and dark purple grids represent the invalid signatures and the blue group of signatures indicates that all signatures in the group are valid, and the detecting branch should be terminated.

The algorithm is described as follows. Firstly, aggregate the input and verify the aggregated signature. If the verification passes, it reveals all are valid, and the function returns an empty set; if not, it will divide the input into two batches. Then, treat each batch as the new input of the algorithm and call the BDD recursively. Once the input contains only one signature, the algorithm should be terminated. Eventually, the algorithm will return a set which exactly includes all invalid signatures.

A randomly selected test was proposed by Guan et al. [14], the approach randomly chooses half of the signature and aggregates them into a batch. Bellare et al. [15] proposed a small exponent test, which also randomly selects 1, 2, 4, . . . signatures in each verification process. It improves the efficiency a little. Later, Hwang et al. [16] simplified the approach. Both are inefficient when we care about the exact invalid signatures.

3. Preliminaries

This section presents the basics related to signature verifications in VANETs. Note that this paper is based on popular and efficient schemes: certificateless signature (CLS) scheme and certificateless aggregate signature (CL-AS) scheme [4, 17]. The notations are listed in Table 1.

3.1. Elliptic Curve Cryptography. As widely used in cryptographic, elliptic curve cryptography (ECC) is an excellent algorithm which has an extremely high efficiency and a relatively excellent security [18, 19]. It can use much fewer bits, less time, and memory to encrypt messages than RSA algorithm in the field of public key cryptography. Thanks to the excellent advantages, ECC can be perfectly applied to the application scenarios of VANETs [20]. We will make the following three definitions to describe ECC.

Definition 1 (elliptic curve definition). Assume that F_p is a finite field of the module p, where p is a large prime number. The elliptic curve over a finite field F_p can be defined as

```
Input: Nonempty Set A, start index i, end index j
Output: Set F
function BDD (A, i, j)
    if Verify (A, i, j) then
        return ∅
    end if
    if i == j then
        return A[i]
    end if
    F = F + BDD(A, i, (i + j)/2)
    F = F + BDD(A, i, (i + j)/2 + 1, j)
    return F
end function
```

Algorithm 1: Binary divisions' detection algorithm.

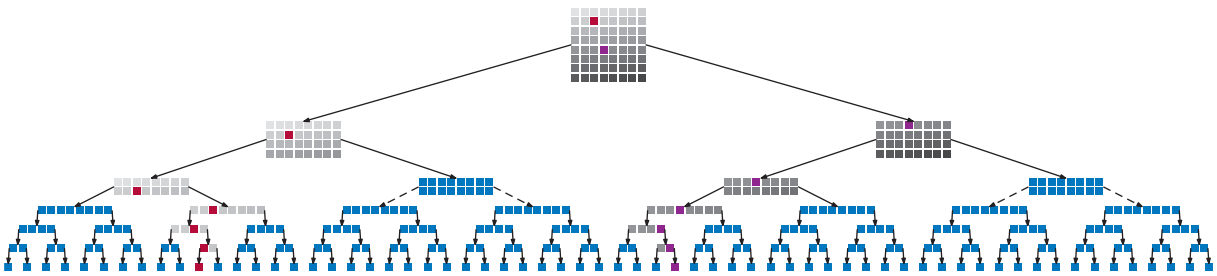


Figure 1: Binary-based aggregate verification process of Huang et al. [8].

Table 1: List of notations.

Notation	Description
TA	Trusted authority
KGC	Key generation center
RSU	Roadside unit
p	A secure prime numbers
E	An elliptic curve: $y^2 = x^3 + ax + b \bmod p$
P	A generator of the additive group
F_p	The finite field of the elliptic curve
$(\mathrm{PK}_{\mathrm{TA}}, \alpha)$	The public key and private key of the TA
$(\mathrm{PK}_{\mathrm{KGC}}, \beta)$	The public key and private key of the KGC
$(\mathrm{PK}_{V_i}, \rho_i)$	The public key and private key of the vehicle
PPK_i	Partial private key of the vehicle V_i
σ_i	The signature of vehicle i
PID_i	Pseudoidentity of the vehicle V_i

E: $y^2 = x^3 + ax + b (\mathrm{mod} p)$, where $a, b, x, y \in F_p$ and $\Delta = 4a^3 + 27b^2 \neq 0 (\mathrm{mod} p)$.

Definition 2 (addition of elliptic curves). Assume that $P = (x_1, y_1) \in E$, where P is a point of the elliptic curve E, and $-P = (x_1, -y_1)(\mathrm{mod} p)$ is the negative point of P. Suppose $Q = (x_2, y_2) \in E, Q \neq -P$, and we can define a line l passes through P and Q and intersects the elliptic curve at a point $R' = (x_3, y_3)$, The symmetrical point about the x-axis with R' is $R = (x_3, -y_3)$; then, we can define $R = P + Q$.

Definition 3 (elliptic curve discrete logarithm problem). Firstly, the discrete logarithm problem should be defined. If a and b are known, how to get k from the equation $b = a^k \mathrm{mod} p$ is a question. Assume that P_1 is a point on the elliptic curve E on the finite field F_p, and select a random number $k \in Z_p^*$. Then, we can calculate $P_2 = k \cdot P_1$. In this case, it is straightforward to calculate P_2 according to Definition 2. However, given the description of elliptic curve discrete logarithm problem (ECDLP), it is scarcely possible to get k according the above equations.

3.2. System Model. In this section, we will try to describe the system model of VANETs in detail. As shown Figure 2, there are four main participants: trusted authority (TA), key generation center (KGC), roadside unit (RSU), and vehicle, which can be divided into two layers: the upper layer includes TA and KGC, both are of more powerful computation, and the lower layer consists of RSUs and vehicles. The demonstration of each participant is as follows:

TA: it is a fully trusted third party that is responsible for system initialization, user registration, system parameter generation, and system security implementation. If necessary, it can track malicious behavior and catch malicious nodes. In addition, it also has enough computing power and storage capacity.

KGC: it is a partially trusted party used for generating partial private key. It can help vehicle generate partial secret key which contributes to its privacy security. It also has sufficient memory, processing, and computing capabilities the same as TA.

Figure 2: System model of VANETs.

RSU: it is a smart application device installed on the roadside, which is able to transmit and submit information to TA, KGC, vehicles, or other RSUs in a secure wired connection and plays the role of an intermediary agent. Unlike TA or KGC, RSU usually has limited computing power and storage capacity.

Vehicle: it is the major and basic member in VANETs, which is generally equipped with a smart device which can perform the basic functions such as transmitting the vehicle's messages and performing a simple calculation. In addition, vehicles commonly have limited computing power and storage capacity. The most obvious feature of the vehicle is that the network connection between vehicles is always in dynamic change.

Note that TA and KGC are distinct entities in function and logical; however, the two usually are deployed on a single server in physics if the scale of VANETs is small.

3.3. Certificateless Aggregate Signature Scheme. Generally, a CLS scheme and a CL-AS scheme consist of the following seven algorithms [21, 22].

Setup: the KGC and TA will execute a probabilistic algorithm, which needs a security parameter λ. Then, it generates an elliptic curve E, the public key $\mathrm{PK}_{\mathrm{TA}}$ and $\mathrm{PK}_{\mathrm{KGC}}$, and the master secret key α and β, respectively. After the security parameters for ECC are generated, a number of system parameters which is used for ensuring the system in order are transmitted between each participant.

ParitialPrivateKeyGeneration: in this algorithm, firstly, the vehicle V_i transmits a tuple, which consists of a real identity and a partial pseudoidentity, to TA. Then, the TA generates a complete pseudoidentity with the tuple and sends it to the KGC for calculation. Eventually, the KGC transmits the partial private key to the vehicle V_i in a secure channel [23].

VehicleKeyGeneration: the vehicle V_i selects a random $\rho_i \in Z_q^*$ as its secret key and calculates its public key $\mathrm{PK}_{V_i} = \rho_i \cdot P$.

IndividualSign: this algorithm is used by each vehicle V_i; after generating a message m_i, the vehicle V_i tries to calculate a set of variables. Then, it sends the signature σ to the verifier.

IndividualVerify: this algorithm is executed by the verifier such as RSU. When the inputs, including signature σ_i, pseudoidentity PID_i, and current time T_{cur}, are received, the verifier will check the validity of the timestamp firstly. Then, the algorithm will output true if the signature is valid or false otherwise.

AggregateSign: in this algorithm, generally, the aggregate signature generator is RSU in the system. For an aggregating set V of n vehicles $V_0, V_1, \ldots, V_{n-1}$, the pseudoidentity PID_i of each vehicle V_i is the list PID, the corresponding public key PK_{V_i} of V_i, and message signature tuples $((m_1, \sigma_1), (m_2, \sigma_2), \ldots, (m_n, \sigma_n))$ of V_i,

respectively. The aggregate signature generator will generate signature σ; then, it will transmit the tuple including the signature, PID, and timestamp list T to the verifier.

AggregateVerify: in general, this algorithm is executed by another RSU. It takes an aggregated set V of n vehicles $\{V_0, V_1, \ldots, V_{n-1}\}$, the pseudoidentity PID_i of each vehicle V_i. The verifier will check the timestamp validity for each vehicle firstly. Then, it will output true if the signature is valid or false otherwise.

To demonstrate the whole communication process, Figure 3 shows the certificateless aggregate signature scheme.

As shown in Figure 3, there are two main steps for CL-AS, aggregation and verification, which cause some delays in VANETs. For convenience, the delay brought by the aggregation of signatures is defined as T_a, while T_c is the total time in a cryptographic operation of the verification process. Without loss of generality, in the following algorithm, the tokens, rather than the actual number, will be used in the relevant calculation because different schemes adopt different algorithms and different time costs. T_a and T_c of different VENETs schemes are shown in Table 2 for reference. From the table, we can conclude that T_a is much smaller than T_c in whichever scheme. Even in some schemes, T_c is almost 1000 times of T_a. Therefore, in the following sections, T_a can be ignored and is not calculated.

3.4. Problem Definition. Batch verification is widely used in VANETs [5, 9, 26]. There are two major research aspects about the batch verification: one is about the correctness and security of the encryption algorithm and how to aggregate the signatures into one and verify the aggregated one correctly and the other is about the strategy of grouping. An appropriate strategy will benefit the aggregation and verification process with significant time savings.

This paper focuses on the latter one, the strategy of grouping, that is, how to divide all signatures into one or more batches to determine all invalid signatures as soon as possible. Here, the problem can be converted into a logical problem, given an input set whose size is $n\,(n \geq 1, n \in \mathbb{Z}^+)$. In the set, all n elements are either true or false, and assume that they are distributed randomly and testing an element is true or false costs much more time than other operations, such as AND. The number of false elements is x, and the number of true elements is $n - x$. The problem is how can we design an algorithm to determine exactly all false elements as efficiently as possible.

In summary, the problem is as follows: given a signatures' set $A = \{\sigma_1, \sigma_2, , \sigma_3, \ldots, \sigma_n\}$ which includes a certain number of valid signatures and invalid signatures, find a set F only including all invalid signatures, i.e., $A - F$ is verified true and minimized to $|F|$.

4. The Proposed Bitwise Divisions' Algorithm

In this section, factorial bitwise divisions' (FBD) algorithm and early-stopping factorial bitwise divisions' (EFBD) algorithm will be fully introduced. In addition, three theorems will be proved to support FBD and EFBD.

4.1. Description of FBD. FBD is a recursive algorithm, and the maximal recursion time is $\log_2 n$. Note that the output set F equals to the input set A in the beginning. First of all, it will check whether all signatures are valid or not. If so, it will return the full set F, which means all signatures in the input set A are valid. And if the size of the set A is 1, it means that the algorithm has gone into the final layer and the algorithm will return (Algorithm 2).

Figure 4 shows the FBD algorithm flow which selects a sample to demonstrate the entire process. Function DivideByBits is utilized to divide the set A to $2\log_2 n$ groups. It is a bitwise operation. For example, a signature whose index is 12 and the binary representation is 001010. Therefore, it is expected to be divided into groups $S_1^0, S_2^1, S_3^0, S_4^1, S_5^0, S_6^0$. The superscript represents that the group is divided by 0 or 1, and the subscript represents which bit is used to divide the set. All groups in G are verified and the set X includes all invalid groups. Then, we can filter F and make it smaller. Recursive boundary is the size of F is 1 or 0. If no signatures are filtered, then the algorithm randomly selects a certain bit and gets two original groups S_i^0, S_i^1 and reorders them and makes them as input to recursively call the algorithm. If some signatures are filtered, then use the output set F as an input to recursively call the algorithm until the filter does not work.

A few proofs will be necessary to demonstrate to support FBD and EFBD algorithms. Theorems 1 and 2 prove the recursive boundary, and Theorem 3 proves that the result is correct. With the following four proofs, FBD and EFBD can work normally and efficiently.

Theorem 1. *If the output set F is an empty set, then the algorithm is supposed to be terminated.*

Proof. As described in the algorithm, if the output set F is an empty set, then all signatures are filtered and there exist no invalid signatures in set A. Therefore, the original proposition is correct. □

Theorem 2. *If the output set F has just one signature, then the algorithm is supposed to be terminated.*

Proof. As the algorithm goes, if the output set F has only one invalid signature, it means that the number of both invalid and valid groups definitely is $\log_2 n$. According to FBD, one valid group can half the size of F and $\log_2 n$ valid groups can cut the size of F to 1. Therefore, the algorithm can return the output set F and the original proposition is true. □

Theorem 3. *The output set F consists of all invalid signatures.*

Proof. Assume that the output set F does not consist of all invalid signatures, which means that the algorithm has omitted certain invalid signatures. According to the

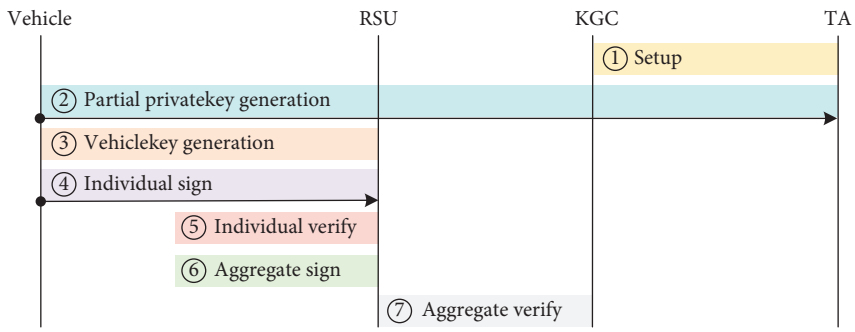


Figure 3: Certificateless aggregate signature scheme.

Table 2: Time cost of different schemes.

Scheme	T_a (ms)	T_c (ms)
Cui et al. [4]	0.0018	1.3298
Kamil et al. [5]	0.0699	2.2899
Kumar et al. [24]	0.2457	4.5120
Malhi et al. [25]	5.2061	12.6330
Zhao et al. [26]	0.0424	17.7673

```
Input: nonempty Set A
Output: set F
function FBD(A)
    if A · size = 1 then
        if Verify (A) then
            return ∅
        else
            return A
        end if
    end if
    G{S_0^0, S_0^1, S_1^0, S_1^1, ...} = DivideByBits(A)
    X = Verify(G)
    F = F − X
    if F · size = 1 then
        return F
    end if
    if F = ∅ then
        return ∅
    end if
    if F · size = A · size then
        choose two groups S_i^1, S_i^0 on random bit
        F = F − FBD(Reorder(S_i^0))
        F = F − FBD(Reorder(S_i^1))
    else
        F = F − FBD(F)
    end if
    return F
end function
```

Algorithm 2: Factorial bitwise divisions' algorithm.

algorithm, the only way to filter from the set F is that the group is verified as true. However, when there exist invalid signatures in a certain group, it will definitely not pass the verification, and the whole group will not be filtered which contradicts the assumption. Therefore, the original proposition is correct. □

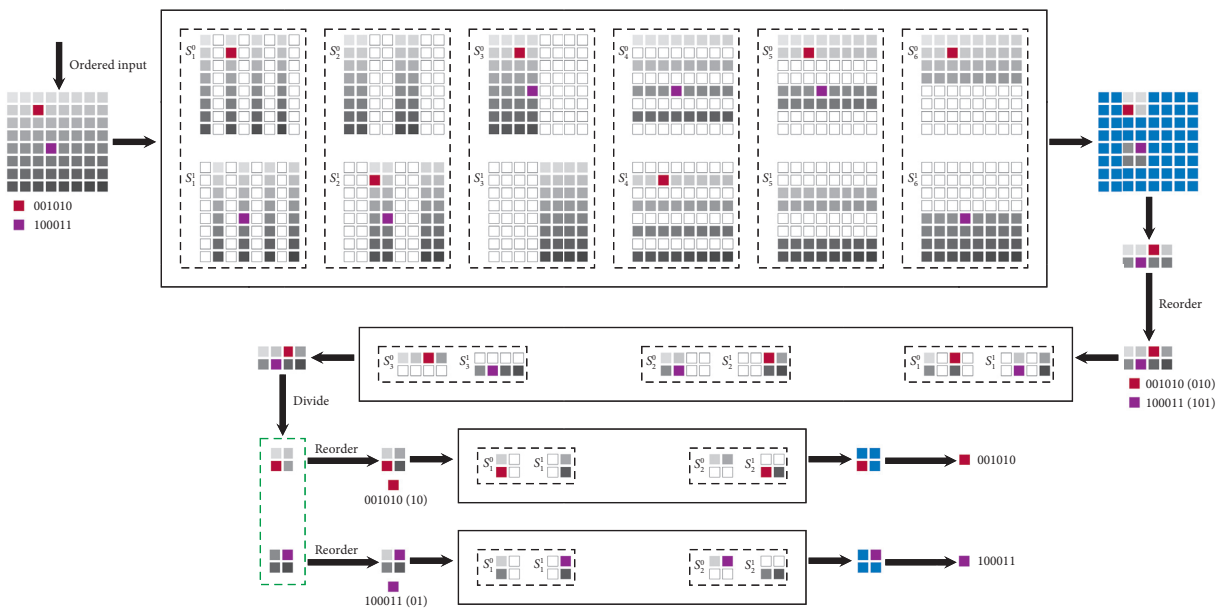


Figure 4: Proposed factorial bitwise divisions' algorithm flow.

Theorem 4. *In the output set F, if the number of invalid signatures is greater than that of valid signatures, then the scale of F will not be reduced.*

Proof. Assume that the number of input set A is $n, n = 2^k$, the number of invalid signatures is $x, x > n/2$, and the number of valid signatures is $n - x$ which is small than $n/2$. In the grouping of certain bit i, S_i^0 and S_i^1 can be seen as two boxes whose capacity is $n/2$. When the number of invalid signatures is greater than the capacity of one box, the other box surely has at least one invalid signature whatever the distribution is. Since both boxes have at least one invalid signature, then they will both be verified failed. Also, the inference is true for other bits. Eventually, all groups are false and no more signatures can be removed from the output set F. Therefore, the original proposition is correct.

Consider its contrapositive. In the output set F, if the scale of F can be reduced, then the number of valid signatures is greater than that of invalid signatures. Indeed it is true. However, it is converse. In the output set F, if the number of valid signatures is greater than that of invalid signatures, then the scale of F which can be reduced is obviously false. Unfortunately, it is exactly what FBD needs. It can easily find its counterexample which can be called repulsion number.

The filter will definitely be invalid and cannot filter any valid signatures. Then, the algorithm will reorder each group. The group can go back to FBD as the group A recursively. Eventually, the algorithm will go to the recursive boundary which can be seemed as a single verification. □

4.2. Description of EFBD. In FBD, it is usual that the majority of the output set F are invalid signatures. As FBD goes, it will branch to the final layer which is a waste of time. As an optimized algorithm for FBD, EFBD adopts single verification when the size of the set F is less than $2 \log_2 n$. The algorithm is designed specially for parallel calculation, and $2 \log_2 n$ verifications can be calculated at a time. The specific algorithm is shown in Algorithm 3.

Most part of EFBD is completely consistent with FBD. The difference is that when the loop terminates, it will singly check each signature in F which will filter all valid signatures. Eventually, it will get a final set F including exactly all invalid signatures.

Definition 4. For two or more binary codes, if both 0 and 1 appear in every bit, then they are called repulsion numbers.

If there exist false repulsion numbers, as EFBD algorithm goes, they will be divided into different groups no matter which bit is used to group. As a result, all groups will be verified false and the filter will be invalid.

5. Performance Analysis

In this section, the performance and time complexity will be analyzed. In addition, our discussion is based on a parallel calculation. For convenience, we define $N = 2 \log_2 n$ which equals to the number of groups in a bitwise partition. For BDD algorithm, it can calculate at most N nodes in the same layer at a time. However, it cannot execute cross-layer computing. For FBD and EFBD, they can also calculate N nodes at a time.

If there exist no invalid signatures, then both algorithms need only one verification. Then, one invalid signature circumstance can be idealized. Time complexity of binary search is reproduced below:

$$O(n) = \left\lceil \frac{1}{N} \right\rceil + \underbrace{\left\lceil \frac{2}{N} \right\rceil + \left\lceil \frac{2}{N} \right\rceil + \cdots + \left\lceil \frac{2}{N} \right\rceil}_{\log_2 n \text{ items}} = 1 + \log_2 n \times \left\lceil \frac{2}{N} \right\rceil. \tag{1}$$

```
Input: nonempty Set A
Output: set F
function EFBD (A)
    if A · size = 1 then
        if Verify (A) then
            return ∅
        else
            return A
        end if
    end if
    G{S_0^0, S_0^1, S_1^0, S_1^1, ...} = DivideByBits (A)
    X = Verify (G)
    F = F − X
    if F · size = 1 then
        return F
    end if
    if F = ∅ then
        return ∅
    end if
    if F · size <= 2 log_2 n then
        return SingleVerify (F)
    end if
    if F · size = A · size then
        choose two groups S_i^1, S_i^0 on random bit
        F = F − EFBD (Reorder (S_i^0))
        F = F − EFBD (Reorder (S_i^1))
    else
        F = F − EFBD (F)
    end if

    return F
end function
```

Algorithm 3: Early stopping bitwise divisions' algorithm.

In general, N is always greater than or equal to 2. Therefore, equation (1) can be further simplified:

$$O(n) = 1 + \log_2 n. \tag{2}$$

According to the algorithm, the time complexity of FBD is $O(n) = (2 \log_2 n/N) = 1$.

Considering the worst case, that is, all signatures are invalid; BDD algorithm will branch on each leaf. Its time complexity is as follows:

$$\begin{aligned}(n) &= \left\lceil \frac{1}{N} \right\rceil + \left\lceil \frac{2}{N} \right\rceil + \left\lceil \frac{4}{N} \right\rceil + \cdots + \left\lceil \frac{n}{N} \right\rceil \\ &\geq \frac{1}{N} + \frac{2}{N} + \frac{4}{N} + \cdots + \frac{n}{N} \\ &\geq \frac{2n-1}{N}.\end{aligned} \tag{3}$$

Time complexity of FBD is as follows:

$$\begin{aligned}O(n) &= \left\lceil \frac{2 \log_2 n}{N} \right\rceil + \left\lceil \frac{2^2 (\log_2 n - 1)}{N} \right\rceil + \left\lceil \frac{2^3 (\log_2 n - 2)}{N} \right\rceil \\ &\quad + \cdots + \left\lceil \frac{2^{\log_2 n} [\log_2 n - (\log_2 n - 1)]}{N} \right\rceil \\ &= \left(\left\lceil \frac{2 \log_2 n}{N} \right\rceil + \left\lceil \frac{2^2 \log_2 n}{N} \right\rceil + \left\lceil \frac{2^3 \log_2 n}{N} \right\rceil + \cdots + \left\lceil \frac{2^{\log_2 n} \log_2 n}{N} \right\rceil \right) \\ &\quad - \left(\left\lceil \frac{2^2}{N} \right\rceil + \left\lceil \frac{2^3 \times 2}{N} \right\rceil + \left\lceil \frac{2^4 \times 3}{N} \right\rceil + \cdots + \left\lceil \frac{2^{\log_2 n} (\log_2 n - 1)}{N} \right\rceil \right) \\ &\leq \left(1 + 2 + 4 + \cdots + 2^{\log_2 n - 1}\right) - \frac{2n \log_2 n - 4n + 4}{N} \\ &= \frac{4n - 4 - 2 \log_2 n}{N}.\end{aligned} \tag{4}$$

Table 3: Number of T_c used in different algorithms.

	BDD					FBD					EFBD				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
4	3	3	3	3	3	1	3	3	4	4	1	1	1	1	1
8	4	4	4	4	4	1	3	4	5	6	1	2	2	3	3
16	5	5	5	5	5	1	4	5	6	8	1	2	2	3	3
32	6	6	6	6	6	1	4	6	8	10	1	2	4	4	5
64	7	7	7	7	7	1	5	8	9	11	1	3	4	6	7
128	8	8	8	8	8	1	5	8	11	13	1	3	5	7	8

Table 4: Number of T_a used in different algorithms.

	BDD					FBD					EFBD				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
4	2	2	2	2	2	1	1	1	1	1	3	3	3	3	3
8	3	3	3	3	3	3	4	4	4	4	3	6	7	8	8
16	4	4	4	4	4	7	9	11	12	13	7	14	16	19	20
32	7	7	8	8	8	15	19	25	30	32	15	31	34	37	42
64	11	12	13	13	14	31	38	56	66	72	31	59	66	79	87
128	20	23	24	26	27	63	77	108	140	155	63	115	123	155	177

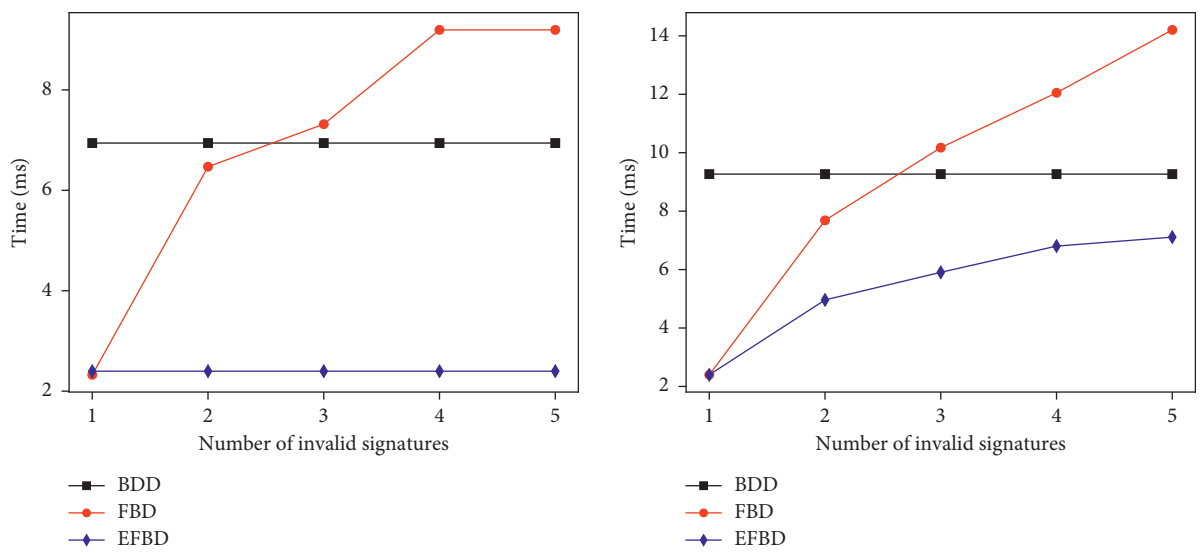


Figure 5: Continued.

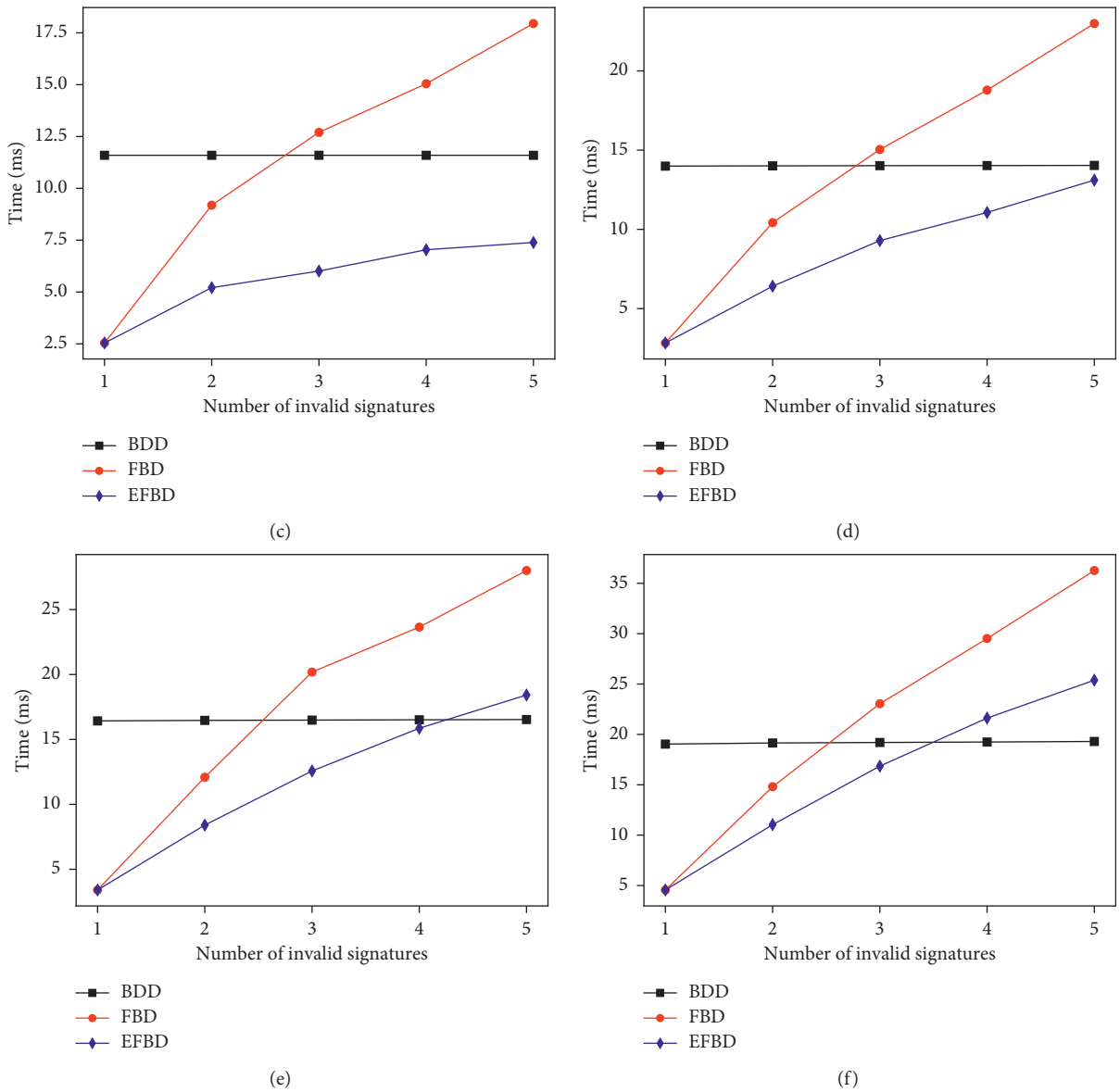


Figure 5: Total time cost of different number of vehicles. (a) $n = 4$. (b) $n = 8$. (c) $n = 16$. (d) $n = 32$. (e) $n = 64$. (f) $n = 128$.

6. Simulation Analysis

In the simulation experiment, BDD, FBD, and EFBD are simulated to count T_a and T_c, respectively, in different scales and invalid signatures. In addition, the parallel calculation is adopted and the scale $N = 2 \log_2 n$.

For different invalid signatures and scales, the performance of the three algorithms is shown in Tables 3 and 4. The number of signatures is from 4 (2^2) to 128 (2^7) and the invalid signatures are 1, 2, 3, 4, and 5, respectively. Because the distribution of invalid signatures greatly influences the result, the simulation is repeated 1000 times and the result is equilibrated to avoid randomness.

As shown in Table 3, when the number of signatures is 4, EFBD only uses $1T_c$ for the reason that it will use single verification when the size of the input set is less than or equal to $N = 2 \log_2 n$. In addition, when the number of false nodes is 1, FBD and EFBD only need $1T_c$ because of their unique filtering structure. BDD has a relatively constant number because of its regular hierarchic execution. In comparison, FBD and EFBD have a high performance when the number of invalid signatures is less than 2 and 4, respectively. BDD has to go through $\log_2 n$ layers to find the invalid signatures. However, as described in Section 4, FBD and EFBD can be seen as a filter, and it can efficiently filter most of the valid signatures and get a set containing invalid signatures when there exist a few invalid signatures. In Table 4, although the number of T_a of our algorithms is bigger than that of BDD, T_a is defined as a much smaller time cost compared to T_c, which makes no difference in the circumstance.

Furthermore, Figure 5 shows the total time cost comparison among the three algorithms. It should be noted that the total time cost includes both aggregation time and verification time. In our work, T_a is 0.036 ms, and T_c is

2.2899 ms which is much bigger than T_a. Therefore, in Figure 5, the rank of different algorithms is the same as the rank in Table 3. From the figures, when the number of elements and the number of false elements are small, the two proposed algorithms are better than the BDD significantly. Actually, in a real scene, the number of elements and the number of false elements is small. Therefore, FBD and EFBD algorithm can achieve high efficiency in real application scenarios.

7. Conclusion

To solve the problem that batch verification has low efficiency in the process of message verification, FBD and EFBD are proposed, and they are featured by parallel calculation and filtering. They show their high efficiency after simulation experiments. However, FBD and EFBD have relatively low efficiency performance in serial computing and only have high efficiency in parallel calculation. And, they cannot outperform BDD when invalid signatures are more than a certain number. According to the experiments, when the number of invalid signatures is more than 4, FBD and EFBD have low efficiency. In the future, FBD and EFBD can be simulated in a more real scene and some factors can be considered. Furthermore, they can be implemented in the hardware that supports parallel computing. Due to repulsion number problem, they can also be optimized to attain higher performance.

Data Availability

The data used to support the findings of the study are included within the article.

Disclosure

This paper was presented in Journal of Physics: Conference Series, Volume 1856, 2021 International Conference on Computer Network Security and Software Engineering (CNSSE 2021), February 26–28 2021, Zhuhai, China.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by Key international cooperation projects of the National Natural Science Foundation of China (No. 61520106007).

References

[1] M. M. Zanjireh and H. Larijani, "A survey on centralised and distributed clustering routing algorithms for WSNS," in *Proceedings of the 2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–6, Glasgow, UK, May 2015.

[2] C.-K. Toh, "Future application scenarios for MANET-based intelligent transportation systems," in *Proceedings of the Future Generation Communication and Networking (FGCN 2007)*, Jeju, People Republic of Korea, January 2008.

[3] Y. Zhou, L. Tian, C. Zhu, X. Jin, and Y. Sun, "Video coding optimization for virtual reality 360-degree source," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 1, pp. 118–129, 2020.

[4] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Information Sciences*, vol. 451-452, pp. 1–15, 2018.

[5] I. A. Kamil and S. O. Ogundoyin, "An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks," *Journal of Information Security and Applications*, vol. 44, pp. 184–200, 2019.

[6] C.-H. Chen, F. Song, F.-J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, Article ID 123344, 2020.

[7] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[8] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "Abaka: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 248–262, 2011.

[9] Y.-L. Huang, C.-H. Lin, and F.-Y. Leu, "Verification of a batch of bad signatures by using the matrix-detection algorithm," in *Proceedings of the 2011 First International Conference on Data Compression, Communications and Processing*, pp. 299–306, Palinuro, Italy, June 2011.

[10] H. Ferng, J. Chen, M. Lotfolahi, Y. Tseng, and S. Zhang, "Messages classification and dynamic batch verification scheme for vanets," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 1156–1172, 2021.

[11] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[12] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.

[13] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[14] D. J. Guan, E. S. Zhuang, I. C. Chung, and Y.-S. Lin, "Performance analysis of some batch verification methods of digital signatures," in *Proceedings of the 2017 12th Asia Joint Conference on Information Security (AsiaJCIS)*, pp. 10–14, Seoul, People Republis of Korea, August 2017.

[15] M. Bellare, J. A. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, vol. 1403, pp. 236–250, Espoo, Finland, May 1998.

[16] J. Y. Hwang, B. Song, D. Choi, S.-H. Jin, H. S. Cho, and M.-K. Lee, "Simplified small exponent test for batch verification," *Theoretical Computer Science*, vol. 662, pp. 48–58, 2017.

[17] X. Ye, G. Xu, X. Cheng, Y. Li, and Z. Qin, "Certificateless-based anonymous authentication and aggregate signature scheme for vehicular ad hoc networks," *Wireless*

Communications and Mobile Computing, vol. 2021, Article ID 6677137, 16 pages, 2021.

[18] E. P. Quiroz, A. Cuno, W. R. Lovón, and E. Cruzado, "ECC usage on X.509 digital certificates," in *Proceedings of the 2020 IEEE Engineering International Research Conference (EIRCON)*, pp. 1–4, Lima, Peru, Octomber 2020.

[19] R. Yadav, S. Srinivasan, and S. Gupta, "Security analysis of RSA and ECC in mobile wimax," in *Proceedings of the 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pp. 1725–1729, Paralakhemundi, India, Octomber 2016.

[20] Y. Yang, D. He, H. Wang, and L. Zhou, "An efficient blockchain-based batch verification scheme for vehicular ad hoc networks," *Transactions on Emerging Telecommunications Technologies*, 2019.

[21] X. Hu, W. Tan, C. Ma, F. Chen, and C. Yu, "Study on security analysis and efficient improvement of certificateless aggregate signature scheme," in *Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 343–346, Beijing, China, Octomber 2020.

[22] C. Hu, S. Wangan, and Z. Bing, "Certificateless aggregate signature scheme," in *Proceedings of the 2010 International Conference on E-Business and E-Government*, pp. 3790–3793, Guangzhou, China, May 2010.

[23] M. Syfullah and J. M.-Y. Lim, "Data broadcasting on cloud-vanet for IEEE 802.11p and LTE hybrid vanet architectures," in *Proceedings of the 2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)*, pp. 1–6, Ghaziabad, India, Feburary 2017.

[24] P. Kumar, S. Kumari, V. Sharma, X. Li, A. K. Sangaiah, and S. H. Islam, "Secure cls and cl-as schemes designed for vanets," *The Journal of Supercomputing*, vol. 75, no. 6, pp. 3076–3098, 2019.

[25] A. K. Malhi and S. Batra, "An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks," *Discrete Mathematics & Theoretical Computer Ence*, vol. 17, no. 1, 2015.

[26] Y. Zhao, Y. Hou, L. Wang, S. Kumari, M. K. Khan, and H. Xiong, "An efficient certificateless aggregate signature scheme for the internet of vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, pp. 3708–3711, 2020.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5553069, 16 pages
https://doi.org/10.1155/2021/5553069

Research Article

Optimization Modeling and Empirical Research on Gasoline Octane Loss Based on Data Analysis

Ji Guo,[1,2] **Yujia Lou**,[1] **Wanyi Wang**,[1] **and Xianhua Wu**[1,2]

[1]*School of Economics and Management, Shanghai Maritime University, Shanghai 201306, China*
[2]*Collaborative Innovation Center on Climate and Meteorological Disasters, Nanjing University of Information Science & Technology, Nanjing 210044, China*

Correspondence should be addressed to Xianhua Wu; 185390@shmtu.edu.cn

Received 11 February 2021; Accepted 29 April 2021; Published 12 May 2021

Academic Editor: Chi-Hua Chen

Gasoline is one of the most consumed light petroleum products in transportation and other industries. This paper proposes a method for optimizing gasoline octane loss using data analysis technology aimed at optimizing the production process and minimizing the loss of gasoline octane. Firstly, the data are screened and the high-dimensional data are reduced to construct the neural network prediction model optimized by genetic algorithm. After utilizing the model for prediction, the optimal operating condition is achieved. Secondly, ensuring that the gasoline emission meets the standard, the octane loss is reduced by adjusting the operating variables. Thirdly, actual data are collected and calculated to obtain the main operating variables and their optimal operating conditions of a petrochemical company affecting the catalytic cracking gasoline S-Zorb unit, thus providing companies using S-Zorb units with reference data for optimizing gasoline catalytic cracking processes. Fourthly, the superiority of the proposed method was verified by comparing it with the other methods. This paper intends to contribute to better modeling the progress of gasoline catalytic cracking by adequately considering the impact of multiple factors, improving the quality of refined oil products of chemical enterprises, saving the economic cost of chemical enterprises, and protecting the atmospheric environment.

1. Introduction

As the world's most consumed light petroleum product, gasoline is one of the main fuels for automobiles. In 2018, global vehicle consumption is 9.5 million [1]. In the same year, the automobile industry consumes about 1.04 million ktoe of motor gasoline, while the transportation sector consumes 2.65 million ktoe oil products around the world, the automobile industry accounts for nearly half of the total consumption [2]. The exhaust gas emitted by gasoline combustion has a large negative impact on the atmospheric environment [3–6] and residents' health [7]. In 2018, greenhouse gas emissions from the transportation sector account for about 20% percent of the total global emissions [8]. Therefore, the cleaning of gasoline is an important task for all countries in the world to purify air pollution. An important issue for clean gasoline production is to maintain the octane level of crude oil as its combustion performance index while reducing the sulfur and olefin content of crude oil so as to improve the quality of finished gasoline.

In the analysis and modeling of gasoline catalytic cracking, the high complexity of refining technology and the diversity of equipment have led to the nonlinear relationship between the operating variables of the control equipment and the strong coupling. The previous analysis models have fewer variables [9–13], and the models have higher requirements for raw materials [14–16], resulting in a lag in the response of the optimization process of the model; the refined oil obtained has more impurities and worse combustion performance, causing a certain economic loss. To avoid unnecessary losses, the modeling process needs to be optimized.

This article puts forward a novel idea: based on data analysis, the optimization process of gasoline catalytic

cracking was modeled and actual data were collected to verify the practicability of the idea. This article intends to contribute to better modeling progress of gasoline catalytic cracking by adequately considering the impact of multiple factors, to optimize the production process and minimize the loss of gasoline octane.

This paper has seven parts. Besides the introduction, the second part is a literature review; the third part introduces the research ideas, data sources, and data dimensionality reduction processing process and constructs the prediction model; the fourth part proposes the prediction results; the fifth part analyzes the economic benefits gained after reducing octane loss; the sixth part shows the robust test with the PCA method; and the seventh part gives out the conclusion and the direction of future research.

The research idea of this article is of the following steps:

(1) *Data Collection*. Historical accumulated data of a petrochemical company's S-Zorb device are collected, including operating variable data, raw material data, product data, and catalyst data. After adjusting the data frequency difference, 325 samples are formed.

(2) *Data Cleansing*. The maximum and minimum limit method, the PauTa criterion, and the null value processing method are used to deal with the values, with abnormal data and abnormal measured variables eliminated.

(3) *Data Dimensionality Reduction*. The dimensionalities of the processed data are reduced by the locally linear embedding method. The measured variables are retained according to the displayed weights to ensure that the selected variables are representative. Then, these variables are tested for correlation coefficients to remove redundant variables with too high correlation coefficients to ensure the independence between variables.

(4) *Model Building*. The aforementioned dimensionality reduction variables and octane loss value are employed, a BP neural network model based on the genetic algorithm is established, and the optimal octane loss prediction model is obtained after adjusting the parameters.

(5) *Results Forecast*. Using the optimal octane loss prediction model, after forecasting and weighted average according to the desulfurization standard, the predicted optimal operating conditions of the main operating variables are obtained.

(6) *Benefits Analysis*. With the predicted optimal operating conditions obtained, the cost saved by reducing octane loss is compared with the cost of original operating conditions to clearly show the economic benefits.

(7) *Robust Test*. The PCA method is employed in the step of data dimensionality reduction.

Then, the GA-BP model is also constructed, and the fitting degree is tested for the robust test.

The process is shown in Figure 1.

This article puts forward an application-oriented method. Compared with previous studies, the possible innovations in this article are as follows:

(1) A locally linear embedding dimensionality reduction algorithm is used to reduce the dimensionalities of high-dimensional data. Traditional linear dimensionality reduction methods, such as principal component analysis and factor analysis, are not appropriately suitable for these high-dimensional data. The locally linear embedding used in this paper can map high-dimensional variable data to low-dimensional vector space based on maintaining the relationships among data, to reduce the dimensionalities and optimize the modeling process.

(2) A BP neural network model optimized based on a genetic algorithm is used to predict gasoline octane loss. Previous studies are mostly implemented from the perspective of chemical mechanism, reducing the octane loss of refined oil by adjusting device configuration or chemical process, raw material properties, and so on. On the other hand, for a large number of variables and time-series data provided by the factory, the traditional regression analysis is hard to be applied because the degree of freedom loss is serious. This paper analyzes the impact of operating variables on gasoline octane loss through data analysis technology and provides enterprises with operating conditions to reduce gasoline octane loss from a new perspective and can further optimize the operation plan based on the mechanism of refined oil production.

2. Literature Review

The cleaning of finished products of petroleum resources and the maximization of efficiency are currently problems facing the chemical industry. In terms of the cleaning of finished products, the main indicator is to reduce the sulfur and olefin content in the finished products. In terms of efficiency maximization, the main indicator is the retention of the octane number (Research Octane Number, RON) in the finished product, which is the most important indicator reflecting the combustion performance of gasoline and is also the brand name of gasoline.

The catalytic cracking technology used for refining petroleum is to crack heavy oil through the combined action of heat energy and catalyst to transform it into cracked gas, gasoline, and diesel [17]. During the catalytic cracking process, flue gas desulfurization is used [18–20], with olefins converted into liquefied petroleum gas by-products. The light oil using this technology has high productivity and good octane retention. Further improvement of catalytic cracking technology is conducive to better cleaning of finished products and maximum efficiency.

In the past, research on catalytic cracking technology focused mainly on mechanism models and technical improvements. Salazar et al. [21] proposed a process for upgrading nitrogen-rich and sulfur-rich heavy oil feedstock, which reduces the nitrogen and sulfur content while increasing the octane number. Valla et al. [15] studied the influence of various types of catalytic cracking feedstock on the distribution of

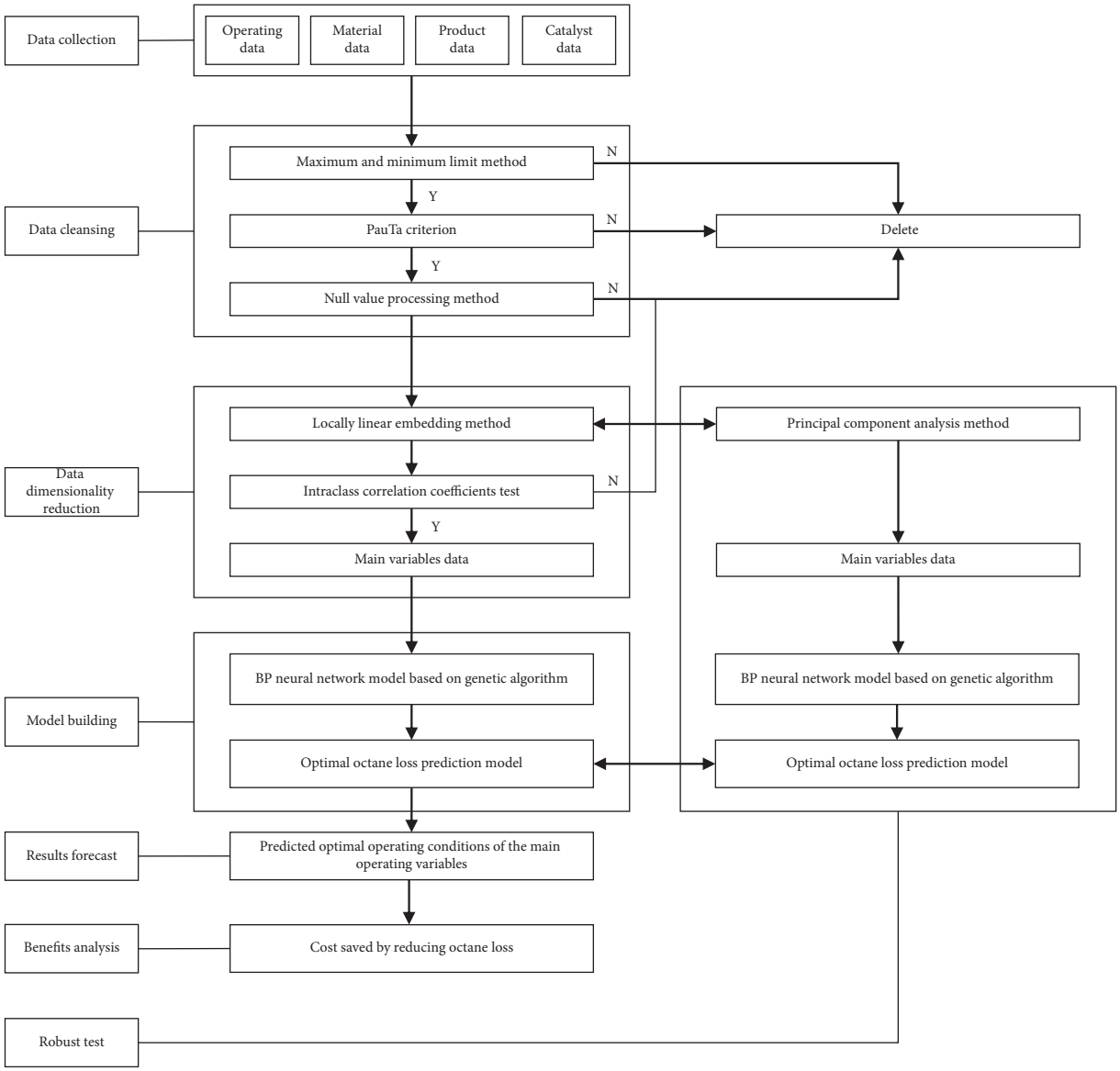


Figure 1: The flow chart of optimization modeling on gasoline octane loss.

sulfides in finished products and proposed the formation mechanism of sulfides on this basis. Brunet et al. [14] investigated the possible sources of sulfur impurities, discussed various factors affecting the hydrodesulfurization and olefin hydrogenation reactions, such as catalysts, carrier properties, and additives, and introduced processes for preserving the octane number of FCC gasoline. Li et al. [22] proposed selective hydrodesulfurization (RSDS-I) technology, which showed the superior desulfurization ability in industrial applications. Li et al. [23] chose CoMoP/η-Al_2O_3 as catalyst to reduce sulfur content of coal tar light oil (CTLO), which is a potential material for the manufacture of high-octane gasoline blending component. Ayoub and Masoud [24] carried out the development of hydrodesulfurization as an alternative for cleaner production of liquefied petroleum gases. Yang et al. [16] analyzed the main factors affecting the change of gasoline octane number, such as the nature of raw materials, catalysts, and device operating conditions. Hasheminejad et al. [25] designed a new material for adsorptive desulfurization to achieve the lowering sulfur level in fuels. Qin et al. [26] established a model for the FCC process on a molecular level employing the structure-oriented lumping (SOL) method to investigate the effects of the diameter expanding reactor. The model established could calculate the molecular level product distribution from the reactor inlet to the outlet to reduce the olefins content and improve the iso-paraffins content of gasoline. Li et al. [27] put forward a high-efficient optimization of corresponding operation conditions for olefin separation. They researched solvent extraction separating olefin and extracted sulfides simultaneously to protect the RON from a loss during hydrodesulfurization.

There are also scholars conducting research on the use of data analysis techniques to establish related models. Qin and Chen [11] proposed a neural network prediction model for gasoline octane number, but only three influencing variables were considered: temperature, pressure, and flow rate of

continuous reforming reactor. Wei [28] used the principal component analysis and BP (back propagation) neural network to process the gas sensor array signal and analyzed the response of the array to gasoline, ethanol, and their mixtures. Yang et al. [13] used a neural network model optimized by the genetic algorithm BP (GA-BP) to establish a gasoline blending model. Paranghooshi et al. [10] used the artificial neural network (ANN) model to determine the octane number of gasoline blends produced by Tabriz refinery. Cheng and Yi [29] proposed a neural network model-based predictive control method for the etherification of FCC light oil. Zhang et al. [30] compared the accuracy of the BP model and the GA-ANN model in predicting the gasoline output of the RFCCU unit. Su et al. [31] used the GA-BP model to predict the production of coke as the main by-product in the catalytic cracking reaction. Ouyang et al. [9] used 19 input variables to establish a BP neural network and studied the influence of raw material preheating temperature, two reaction zone outlet temperatures, and reaction pressure on product distribution. They also used genetic algorithms to optimize the operating variables, and the gasoline yield was significantly improved after optimization. Tian et al. [12] used the particle swarm optimization (PSO-BP) neural network to predict the law of sulfur content in refined diesel products with operating parameters and compared the performance of BP, GA-BP, and PSO-BP. Cheng et al. [32] combined the BP neural network with the PID control, optimized the parameters of the controller, and applied it to the flow control of catalytic cracking natural gas, so as to control the concentration of the final product of heavy oil. Zhang et al. [33] established a nonrandom two-liquid model and a simulation method for the FCC naphtha solvent extraction process to fully improve the structure of the components of FCC naphtha aimed at the production of ultra-low-sulfur gasoline with minimal loss of octane number. Foroughi et al. [34] applied an artificial neural network to predict the volume percentage of adulterants to protect adulteration in gasoline based on the set of an automatic distillation apparatus measuring the recovered volume and temperature. Ma et al. [35] clarified the mechanism of research octane number (RON) and motor octane number (MON) of gasoline influenced by initial thermodynamic conditions and fuel chemistry. They proposed a hybrid analysis framework, with a combination of the transient tracking method and data-driven modeling algorithm to realize the fast-forward prediction and analysis on fuel's ONs.

In summary, in terms of optimizing the process of gasoline catalytic cracking, the academic circles mainly modify the raw material properties and optimize the steps and the equipment in the gasoline catalytic cracking process from the perspective of chemical industry. When using data analysis to adjust the corresponding catalytic cracking operation variables to achieve optimization, most studies selected few variables, and it is difficult to fully consider the impact of multiple factors. This article hopes to contribute in this regard.

3. Data and Methodology

3.1. Data Sources. The original data are the historical accumulated data of the catalytic cracking gasoline S-Zorb unit of a petrochemical company. Operating variable data come from real-time database. The collection time is from April 2017 to May 2020, with a total of 353 operating variable sites collected. From April 2017 to September 2019, the data collection frequency was 3 minutes/time; from October 2019 to May 2020, the data collection frequency was 6 minutes/time. The data time range for raw materials, products, and catalysts was from April 2017 to May 2020. The octane number of raw materials and products is an important modeling variable. Since the octane number is difficult to measure, the data collection frequency is twice a week. The difference in the frequency of data collection is adjusted according to the actual situation. The measured value of the octane number can be regarded as the comprehensive effect within two hours before the moment. The value of the manipulated variable is the average value of the previous two hours, which corresponds to the measured value of the octane number at that moment, resulting in 325 samples with a fixed time interval of two weeks (see Supplemental materials: S-Table 1).

3.2. Data Cleansing. Since the enterprise device has been operating continuously for 4 years, it is necessary to improve the accuracy and effectiveness of the recorded data. Before modeling, the data need to be sorted out. In the original data of the sample, most of the variable data is normal, but some data of each device have problems, some variables only contain data for part of the period, and some of the variables' data are all null values or some of the data are null values. Therefore, this article will process the original data for subsequent research.

First, according to the chemical process requirements and operating experience, the original data variables have a certain operating range (see Supplemental materials: S-Table 2), with the maximum and minimum limiting method used to remove some samples that are not within this range.

Second, outliers are removed according to the PauTa criterion (the 3δcriteria). The variable is measured with equal precision, with $x_1, x_2, \ldots, x_n$ obtained, where$n$ is the number of samples. The arithmetic mean $\overline{x}$ and the residual error $v_i = x_i - \overline{x}$ $(i = 1, 2, \ldots, n)$ are calculated, with the standard error δ determined according to Bessel's formula as follows:

$$\delta = \left(\frac{\sum_{i=1}^{n} v_i^2}{n-1}\right)^{(1/2)} = \left\{\frac{1}{n-1}\cdot\left[\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\right]\right\}^{(1/2)}. \tag{1}$$

If the residual error v_b $(1 \le b \le n)$ of a certain measured value x_b satisfies $|v_b| = |x_b - x| > 3\delta$, x_b is considered to be a bad value with a larger error value and is eliminated.

Finally, the null values in the measured variables of the data sample are processed. After counting, the average of the number of null values in each column is calculated and defined as the critical point for judging incomplete data. If the number of null values in the measured variable exceeds this critical value, it is considered that there are too many

incomplete data in this column of data, which will adversely affect the goodness of fit, so this type of measured variable is eliminated. After calculation, the critical point of incomplete data judgment is 104, and 16 measured variables are eliminated after processing (see Table 1).

3.3. Data Dimensionality Reduction. When modeling, the variables need to be reduced in dimensionality. Dimensionality reduction is conducive to screening out the operating variables having the greatest impact on the octane number in the catalytic cracking process, ignoring secondary variables, and improving application efficiency. Due to the large number of variables, they can be regarded as high-dimensional data; there is a nonlinear relationship between each other; the nonlinear dimensionality reduction algorithm is more suitable for the above situation.

3.3.1. LLE Algorithm. This paper uses locally linear embedding (LLE) [36, 37] in the nonlinear dimensionality reduction algorithm for data dimensionality reduction. Compared with the traditional PCA (principal component analysis) sample variance reduction method, it retains the local linear characteristics of the sample when reducing the dimensionality. The principle of LLE is that high-dimensional data are approximately locally linear in a very small local neighborhood in Euclidean space, and that a certain point can be represented by linear least squares of its surrounding points. LLE uses the linear fitting coefficients as the local geometric properties of the point to find the low-dimensional projection of the data. This algorithm is widely used in the field of high-dimensional data (see Appendix A for the specific algorithm).

3.3.2. Data Dimensionality Reduction Results. After obtaining the weight coefficient matrix through the LLE algorithm, the 29 control variables with the highest weights in the weight coefficient matrix are extracted. Taking into account the actual situation in the chemical process, the "Octane number of raw materials" should be taken as one of the main variables and included in the analysis, resulting in 30 main variables.

The use of the LLE algorithm for dimensionality reduction can ensure that the above main variables are representative and then can determine whether the variables are related by calculating their correlation coefficients. The thermodynamic diagram representing the correlation coefficient of the variable is shown in Figure 2.

It can be seen from Figure 2 that the correlation between some variables and other variables is too high (the absolute value of the correlation coefficient being greater than 0.8), so the seventh, 12th, and 18th variables are deleted. After the deletion, the correlation coefficient of the representative variables decreases (as shown in Figure 3), which ensures the independence between the representative variables.

After processing and testing, 27 variables were finally retained as main variables, with the specific main variables shown in Table 2.

3.4. GA-BP Model Establishment. With the development of intelligent machine algorithms, data analysis techniques can be used to solve the problem of excessively high data dimensions. This paper establishes a GA-BP neural network model to predict the loss of gasoline octane number, aiming to adjust operating variables and reduce the loss of octane number in the chemical process.

3.4.1. Model Design. The back propagation (BP) neural network [38–45] is based on intelligent machine learning, has nonlinear features, good classification ability, and mapping ability to multidimensional functions, and has great advantages in multivariate regression. It has an input layer, an intermediate layer, and an output layer. In essence, the square of the network error is used as the objective function, and the gradient descent method is used to obtain the minimum value of the objective function. The calculation process is divided into two parts: forward propagation and backward propagation. When calculating the error in the forward direction, it goes from the input layer to the output layer, and the reverse adjustment process is to adjust the weight and threshold of the network through the distribution of the error signal in each layer so that the network error decreases along the gradient direction. (Figure 4)

The genetic algorithm (GA) [42, 43] is a parallel random search optimization method based on the theory of biological evolution to simulate the genetic mechanism of nature. Based on the principle of "survival of the fittest" in nature, individuals are selected, crossed, and mutated; individuals are screened according to the selected fitness function, individuals with better fitness are retained, and individuals with poor fitness are eliminated. The new group is better than the previous generation on the basis of inheriting the information of the previous generation. They loop iteratively until the optimization is reached.

As a local search optimization method, the BP neural network is easy to fail in the case of nonlinear complex problems. In addition, the BP algorithm is also prone to overfitting. The genetic algorithm can optimize the neural network learning rules and improve the computational efficiency of the neural network; it is necessary to build a dynamic neural network structure to avoid the final result of the model being a local optimum instead of a global optimum. Therefore, a BP neural network model is employed optimized based on a genetic algorithm, namely, the GA-BP model [44] to predict product octane loss (see Appendix B for specific algorithm design).

3.4.2. Model Results. The input layer X of the neural network is the 27 operating variables screened out above, the output layer P is the octane loss value and the product sulfur content 2 variables, and the formula of the hidden layer node is $Y = \sqrt{n_X + n_P} + L$, $(1 \leq L \leq 10)$, which is set to 7 here. In order to improve the accuracy of neural network prediction, the number of the training set is set to 300 and the number of the test set is set to 25. Each iteration is randomly sampled from the sample set.

Table 1: The eliminated variables.

Order	Variable code	Variable name
1	S-ZORB.FC_2301.PV	D105 fluidized hydrogen flow rate
2	S-ZORB.FT_1501.PV	New hydrogen intake flow rate
3	S-ZORB.FT_5104.PV	Light hydrocarbon discharge flow rate
4	S-ZORB.FT_9101.PV	Oil discharge flow rate
5	S-ZORB.FT_1002.PV	1# catalytic gasoline intake flow rate
6	S-ZORB.FC_1202.PV	D121 top remove flare flow rate
7	S-ZORB.FC_3103.PV	Regenerated cold nitrogen flow rate
8	S-ZORB.FT_1002.TOTAL	S-ZORB.FT_1002cumulative flow rate
9	S-ZORB.FT_1501.TOTAL	New hydrogen intake accumulated flow
10	S-ZORB.FT_5102.PV	S-ZORB.FT_5102 intake flow rate
11	S-ZORB.FT_2901.DACA	D-109 loose wind flow rate
12	S-ZORB.FC_1104.DACA	Feed control valve bypass flow rate
13	S-ZORB.FT_2803.DACA	D-102 emergency hydrogen flow rate
14	S-ZORB.FT_1502.DACA	Supply hydrogen compressor outlet return pipe flow rate
15	S-ZORB.TEX_3103A.DACA	Eh-102 heating element/a beam temperature
16	S-ZORB.FT_5102.DACA. PV	D-201 sulfur-containing effluent discharge flow rate

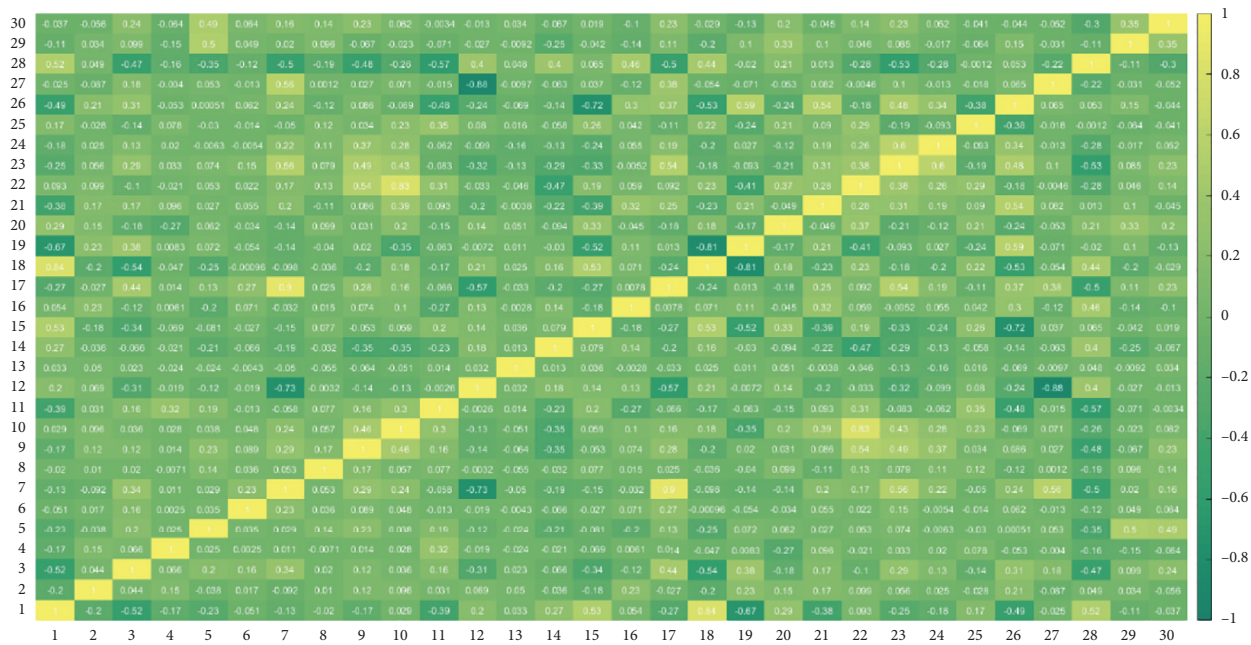


Figure 2: Main variables correlation coefficient thermodynamic diagram before deletion.

After selecting and adjusting the parameters, the genetic algorithm finally sets the parameters as follows: the maximum number of iterations $G_{\max}$ is set to 25, the population size M to 60, the crossover probability P_c to 0.2, and the mutation probability P_m to 0.05. The final parameters of the neural network are set as follows: the learning rate h is 0.1, the minimum error threshold $E_{\min}$ is set to 0.00001, and the maximum number of iterations G_{max} is set to 100.

After model construction, the overall goodness of fit of GA-BP prediction is 52.51%, the prediction error of octane number loss is small, and the prediction error of product sulfur content is large. The error, error percentage, and fitness curve between the predicted value and the true value are shown in Figures 5–7 . The optimized neural network model after training is saved, which is the prediction model of gasoline octane number loss.

It is assumed that the physical and chemical properties of raw materials and standby and regenerated adsorbents in the optimization process remain unchanged (the octane number of raw materials and the sulfur content of raw materials in the main variables also remain unchanged). According to China's current standard GB 18352.6-2016, the sulfur content of finished gasoline products is required to be not more than 10 μg/g. In order to leave room for business operations, the selection criteria for sulfur content in this article are not more than 5 μg/g. For the 25 main operable variables corresponding to the above samples, simulation samples are generated according to the variable range and minimum change value, simulated the debugging process of the variables, and used the octane loss reduction as the weighted average to obtain the optimal operating conditions.

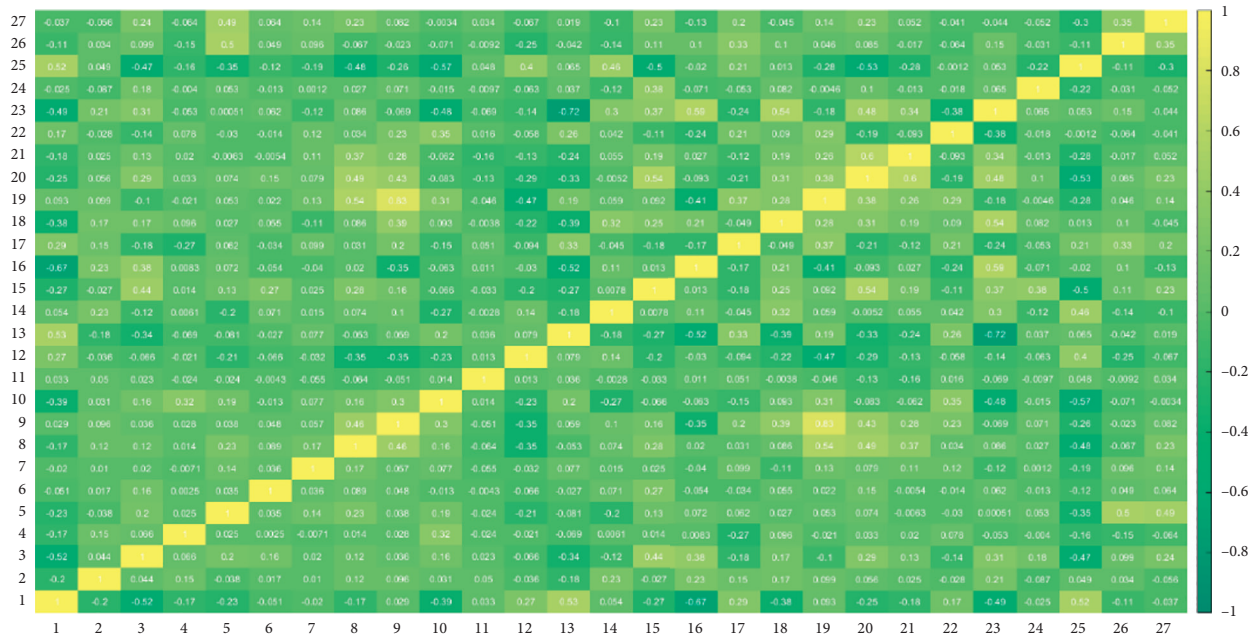


Figure 3: Main variable correlation coefficient thermodynamic diagram after deletion.

Table 2: Main variables after deletion.

Order	Variable code	Variable name
1	S-ZORB.TE_2301.PV	D105 temperature
2	S-ZORB.TE_9301.PV	1.0 MPa steam inlet temperature
3	S-ZORB.FT_3301.PV	Deaerated water inlet flow rate
4	S-ZORB.FT_9401.PV	Purify air inlet flow rate
5	S-ZORB.FT_1004.PV	3# catalytic gasoline intake flow rate
6	S-ZORB.AI_2903.PV	Oxygen content of regenerated flue gas
7	S-ZORB.LI_9102.DACA	D204 liquid level
8	S-ZORB.TE_9002.DACA	D203 top outlet pipe temperature
9	S-ZORB.TE_1502.DACA	D122 outlet pipe temperature
10	S-ZORB.FT_2302.DACA	D105 upper jumper loose wind flow rate
11	S-ZORB.SIS_PT_2602.PV	Regenerator top/regenerator receiver differential pressure
12	S-ZORB.PDT_3602.DACA	Cold nitrogen filter ME-114 differential pressure
13	S-ZORB.FT_3701.DACA	Closed hopper N2 filter outlet gas flow rate
14	S-ZORB.FT_3702.DACA	Lock hopper H2 filter outlet gas flow rate
15	S-ZORB.PDT_2605.DACA	R-102 bottom nozzle differential pressure
16	S-ZORB.PDT_2906.DACA	Me-108 filter differential pressure
17	S-ZORB.TE_7106.DACA	K-101a left exhaust temperature
18	S-ZORB.HIC_2533.AUTOMANA.OP	HV2533 hand operator
19	S-ZORB.TE_5009.DACA	E-205 inlet pipe temperature
20	S-ZORB.AT-0003.DACA.PV	S-ZORB.AT-0003
21	S-ZORB.AT-0010.DACA.PV	S-ZORB.AT-0010
22	S-ZORB.FT_5204.DACA.PV	Degassing of gasoline products
23	S-ZORB.FT_1006.TOTALIZERA.PV	Hydrocracking light naphtha intake accumulated flow
24	S-ZORB.FT_1503.DACA.PV	8.0 MPa hydrogen to circulating hydrogen compressor intake flow rate
25	S-ZORB.FT_1504.TOTALIZERA.PV	8.0 MPa hydrogen to back blow hydrogen compressor outlet accumulated flow
26	RON	Octane number of raw materials
27	S-ZORB.AT_1001.DACA	Sulfur content of raw gasoline

4. Main Results

In summary, we have obtained the main operating variables of the petrochemical enterprise that affect the catalytic cracking gasoline S-Zorb unit, as well as the optimal operating conditions when the product sulfur content is less than 5 μg/g, as shown in Table 3.

The above results provide companies using S-Zorb with reference data for optimizing the process of catalytic cracking gasoline.

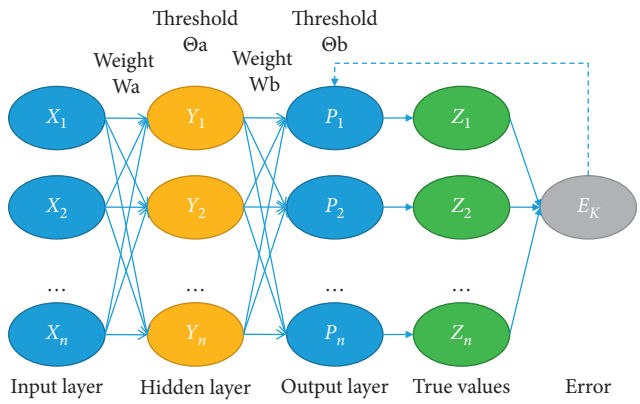


Figure 4: BP neural network structure.

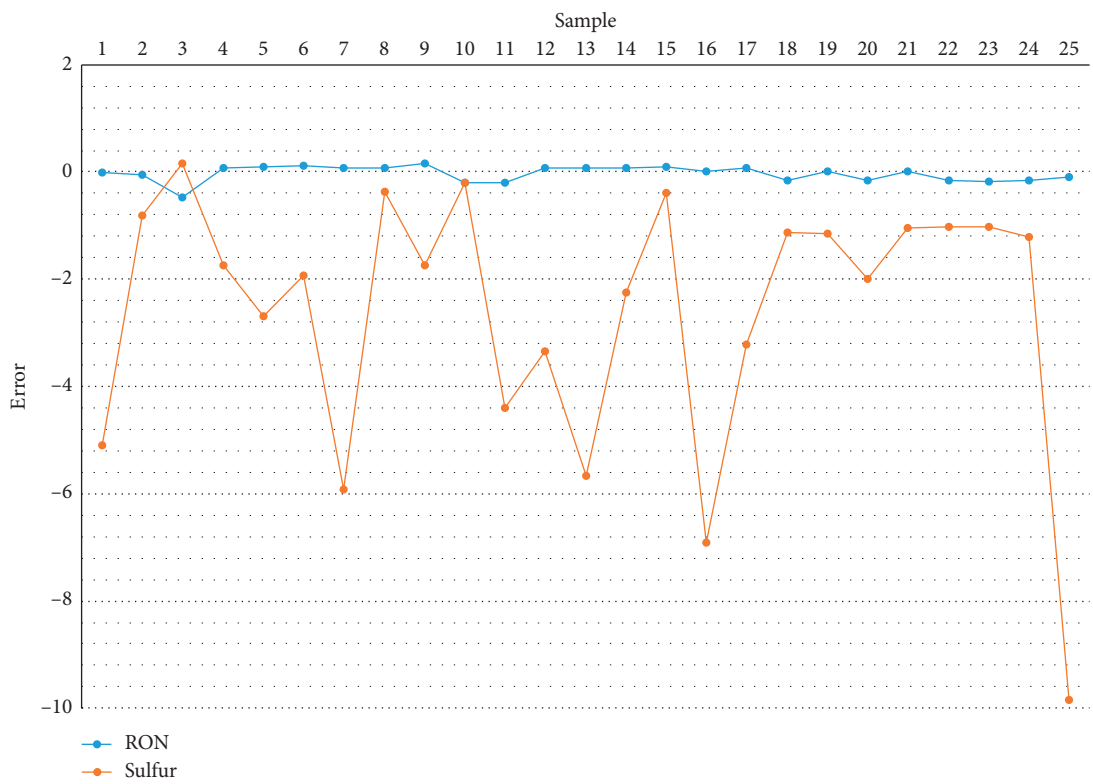


Figure 5: Error between the predicted value and the true value.

5. Economic Benefit Analysis

In the process of refining gasoline with S-Zorb unit in existing petrochemical enterprises, if the octane number is reduced by 1 unit, it is equivalent to reducing the economic loss of $23.055 per ton.

Therefore, in this study, the average RON of the original data sample is 88.45 units, and the average octane number of the product under the optimal operating conditions is 89.70 units. After the prediction of the GA-BP model and the optimization of operating conditions, the octane number loss of the product is reduced by 1.25 units. If we take the automobile industry in 2018 as an example and calculate by the annual gasoline consumption of 1.04 million kt, the application of this research can save about 29.97 million dollars for automobile industry in the whole world.

It can be seen that reducing the octane loss of gasoline products will bring huge economic cost savings to enterprises and will also cut down air pollution and the corresponding cost of treatment, so the conclusions of this study can provide excellent economic benefits in practice.

6. Robust Test

To verify the reliability of the method and conclusion in this paper, the principal component analysis (PCA) [46] method

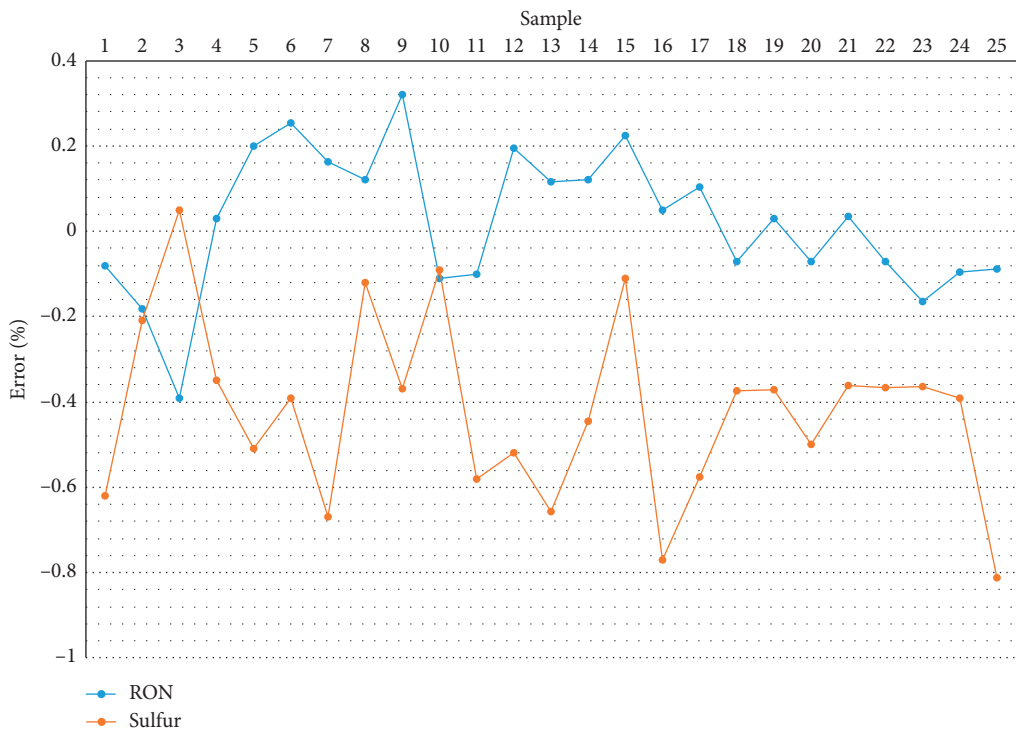


Figure 6: Error percentage between the predicted value and the true value.

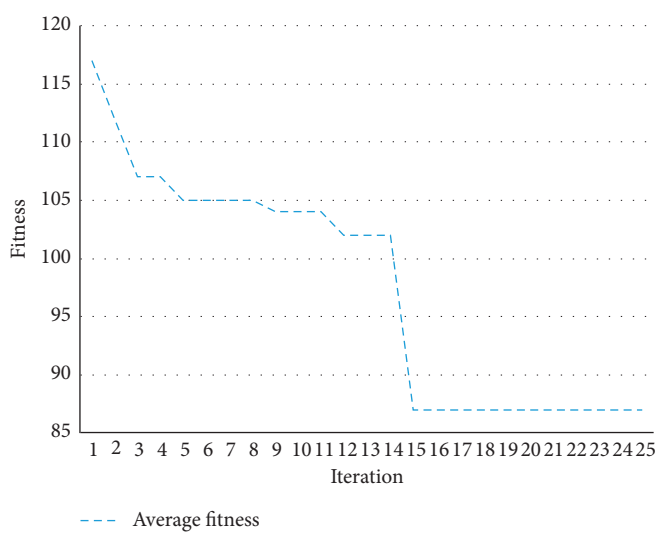


Figure 7: The fitness curve.

is used to reduce the dimension of the original variables and sample data, and then the GA-BP model is also constructed, and its fitting degree is tested for the robust test.

PCA aims at trying to create a new set of unrelated comprehensive indicators composed of the original indicators with different weights. Since it is necessary to adjust the operating variables of the S-Zorb device to achieve the goal of reducing octane number loss, the first 26 variables are also retained according to the weight and the variable "Octane number of raw materials" is added. For the variables retained that are already representative and unrelated, the calculation of their correlation coefficients is omitted. The specific main variables are shown in Table 4.

Then, the main variables above are employed by GA-BP modeling. After model construction, the overall goodness of fit is 45.36% with a similar error performance. The error, error percentage, and fitness curve between the predicted value and the true value are shown in Figures 8–10 .

It can be concluded that the overall fit goodness of the PCA method is inferior to that of the LLE method. For this study focused on more than 300 variables, the PCA method is not appropriately suitable for such high-dimensional data.

Table 3: The optimal operating conditions of the main operating variables.

Order	Variable code	Variable name	Optimal operating conditions	Unit
1	S-ZORB.TE_2301.PV	D105 temperature	277.5374	°C
2	S-ZORB.TE_9301.PV	1.0 MPa steam inlet temperature	194.7160	°C
3	S-ZORB.FT_3301.PV	Deaerated water inlet flow rate	589.8554	kg/h
4	S-ZORB.FT_9401.PV	Purify air inlet flow rate	392.3236	Nm^3/h
5	S-ZORB.FT_1004.PV	3# catalytic gasoline intake flow rate	37.6089	T/h
6	S-ZORB.AI_2903.PV	Oxygen content of regenerated flue gas	0.9389	%
7	S-ZORB.LI_9102.DACA	D204 liquid level	45.7631	%
8	S-ZORB.TE_9002.DACA	D203 top outlet pipe temperature	29.7332	°C
9	S-ZORB.TE_1502.DACA	D122 outlet pipe temperature	44.6834	°C
10	S-ZORB.FT_2302.DACA	D105 upper jumper loose wind flow rate	102.2920	Nm^3/h
11	S-ZORB.SIS_PT_2602.PV	Regenerator top/regenerator receiver differential pressure	-126.1370	kPa
12	S-ZORB.PDT_3602.DACA	Cold nitrogen filter ME-114 differential pressure	0.4967	kPa
13	S-ZORB.FT_3701.DACA	Closed hopper N2 filter outlet gas flow rate	25.7114	Nm^3/h
14	S-ZORB.FT_3702.DACA	Lock hopper H2 filter outlet gas flow rate	25.4971	Nm^3/h
15	S-ZORB.PDT_2605.DACA	R-102 bottom nozzle differential pressure	12.1455	kPa
16	S-ZORB.PDT_2906.DACA	Me-108 filter differential pressure	10.3282	kPa
17	S-ZORB.TE_7106.DACA	K-101a left exhaust temperature	32.1123	°C
18	S-ZORB.HIC_2533.AUTOMANA.OP	HV2533 hand operator	76.8424	°C
19	S-ZORB.TE_5009.DACA	E-205 inlet pipe temperature	27.0397	°C
20	S-ZORB.AT-0003.DACA.PV	S-ZORB.AT-0003	3.0541	—
21	S-ZORB.AT-0010.DACA.PV	S-ZORB.AT-0010	1.0862	—
22	S-ZORB.FT_5204.DACA.PV	Degassing of gasoline products	1075.6240	kg/h
23	S-ZORB.FT_1006.TOTALIZERA.PV	Hydrocracking light naphtha intake accumulated flow	63021560.0446	kg/h
24	S-ZORB.FT_1503.DACA.PV	8.0 MPa hydrogen to circulating hydrogen compressor intake flow rate	2265244.5657	Nm^3/h
25	S-ZORB.FT_1504.TOTALIZERA.PV	8.0 MPa hydrogen to back blow hydrogen compressor outlet accumulated flow	17278642.1169	Nm^3/h

Table 4: Main variables retained by PCA.

Order	Variable code	Variable name
1	S-ZORB.TE_2002.DACA	R-101 bed middle temperature
2	S-ZORB.FT_1002.PV	1# catalytic gasoline intake flow rate
3	S-ZORB.FT_2901.DACA	D-109 loose wind flow rate
4	S-ZORB.FT_1002.TOTAL	S-ZORB.FT_1002 cumulative flow rate
5	S-ZORB.FT_3201.DACA	D-110 top pressure
6	S-ZORB.AT-0008.DACA.PV	S-ZORB.AT-0008
7	S-ZORB.PT_2106.DACA.PV	Back blow pressure
8	S-ZORB.PC_3501.DACA	Emergency hydrogen main
9	S-ZORB.AT-0011.DACA.PV	S-ZORB.AT-0011
10	S-ZORB.AT-0009.DACA.PV	S-ZORB.AT-0009
11	S-ZORB.AT-0007.DACA.PV	S-ZORB.AT-0007
12	S-ZORB.FT_9402.TOTAL	S-ZORB.FT_9402.TOTAL accumulated flow
13	S-ZORB.TXE_2203A.DACA	Eh-101 heating element temperature
14	S-ZORB.LC_5101.PV	Top reflux tank D201 level
15	S-ZORB.FT_5204.TOTALIZERA.PV	Gasoline product degassing accumulated flow
16	S-ZORB.TE_7508.DACA	K-103a exhaust temperature
17	S-ZORB.PT_7103.DACA	K-101a intake pressure
18	S-ZORB.PT_7107.DACA	K-101a exhaust pressure
19	S-ZORB.TC_2201.PV	Eh-101 inlet flow
20	S-ZORB.PT_7510.DACA	K-103a exhaust pressure
21	S-ZORB.SIS_PT_2703	D-110 bottom pressure
22	S-ZORB.FT_1202.TOTAL	S-ZORB.FT_1202.TOTAL accumulated flow
23	S-ZORB.SIS_LT_1001.PV	Material buffer tank liquid level
24	S-ZORB.TE_7102.DACA	K-101a intake temperature
25	S-ZORB.PT_7508.DACA	K-103a inlet pressure
26	S-ZORB.TE_7506.DACA	K-103a inlet temperature
27	RON	Octane number of raw materials

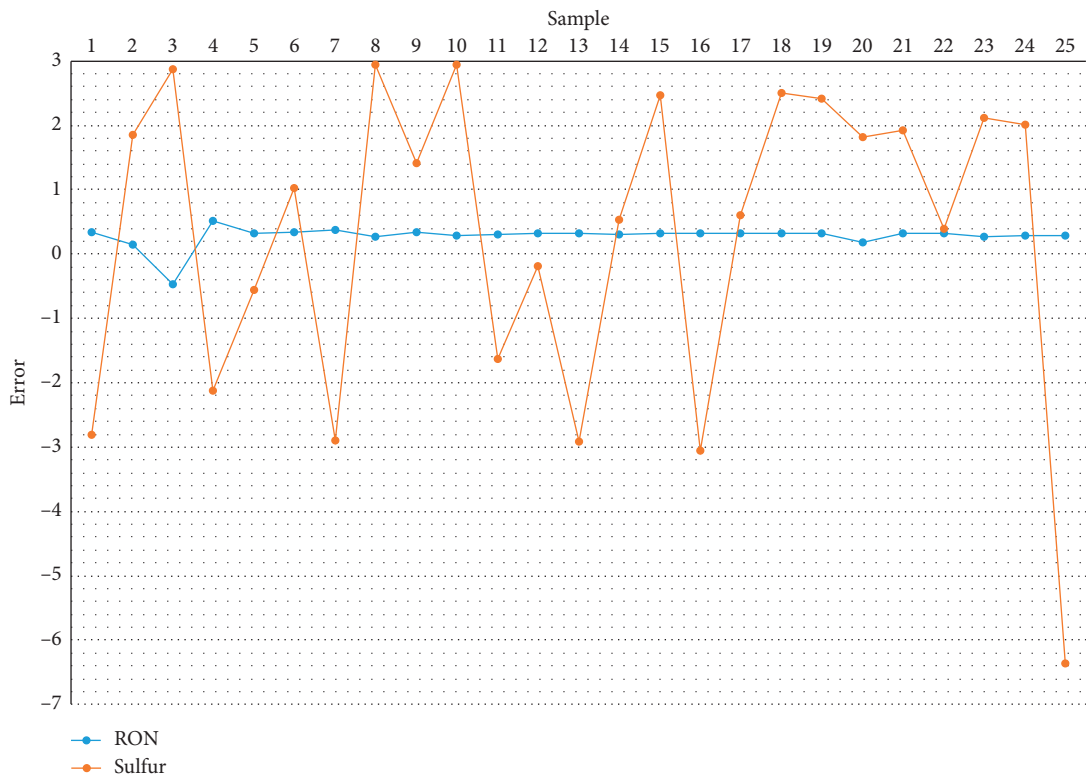


Figure 8: Error between the predicted value and the true value.

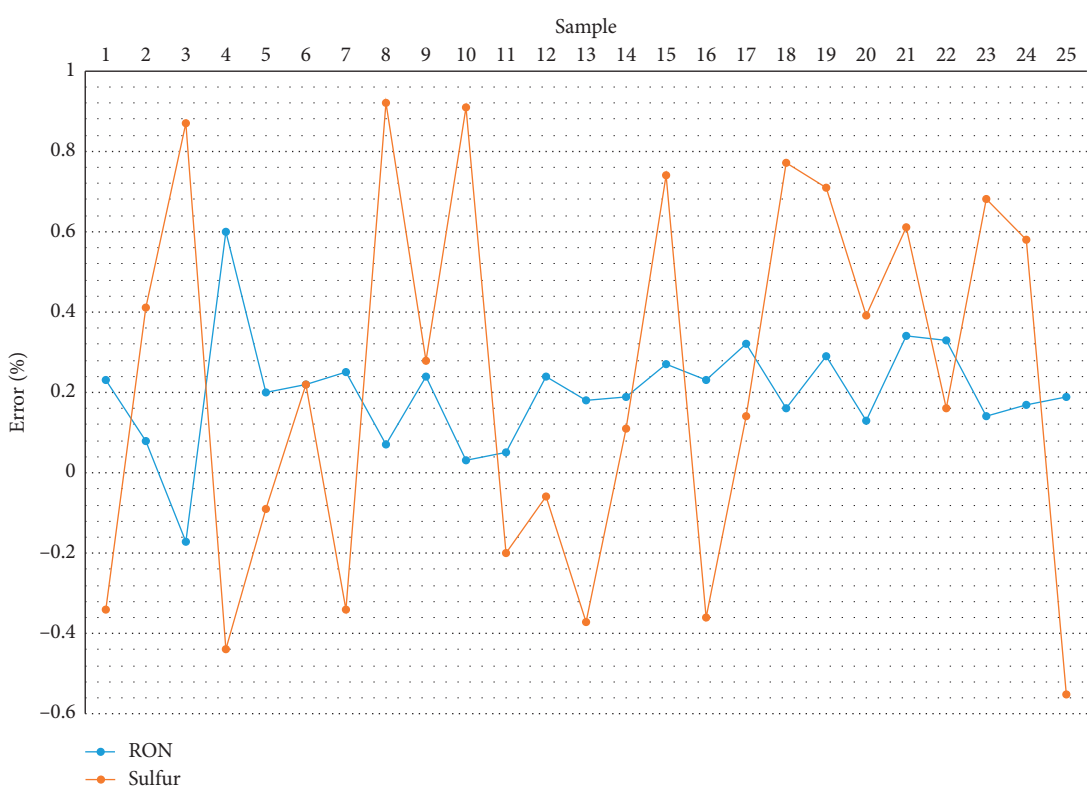


Figure 9: Error percentage between the predicted value and the true value.

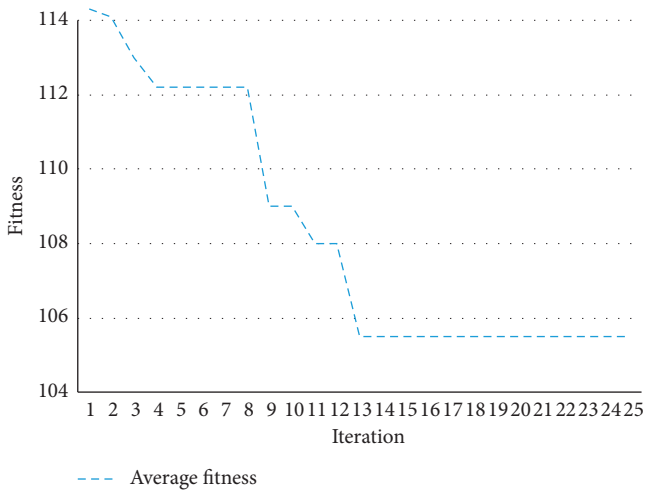


Figure 10: The fitness curve.

7. Conclusion and Discussion

7.1. Conclusion. Automobile traffic is a major source of pollution. Based on data analysis, this paper proposes a new research octane number extraction method to control the harm of automobile pollution from the source. This research result shows the following:

(1) There are 27 main variables artificial remained after dimension reduction by the locally linear embedding method and calculating correlation coefficients. They are D105 temperature, 1.0 MPa steam inlet temperature, deaerated water inlet flow rate, purify air inlet flow rate, 3# catalytic gasoline intake flow rate, the oxygen content of regenerated flue gas, D204 liquid level, D203 top outlet pipe temperature, D122 outlet pipe temperature, D105 upper jumper loose wind flow rate, regenerator top/regenerator receiver differential pressure, cold nitrogen filter ME-114 differential pressure, closed hopper N2 filter outlet gas flow rate, lock hopper H2 filter outlet gas flow rate, R-102 bottom nozzle differential pressure, Me-108 filter differential pressure, K-101a left exhaust temperature, HV2533 hand operator, E-205 inlet pipe temperature, S-ZORB.AT-0003, S-ZORB.AT-0010, degassing of gasoline products, hydrocracking light naphtha intake accumulated flow, 8.0 MPa hydrogen to circulating hydrogen compressor intake flow rate, 8.0 MPa hydrogen to back blow hydrogen compressor outlet accumulated flow, the octane number of raw materials, and sulfur content of raw gasoline.

(2) After employing the GA-BP neural network to obtain the prediction model of gasoline octane number loss, the optimal operating conditions of the main operating variables are achieved (see Table 3) under the condition that product sulfur content is less than 5 μg/g. With the optimal operating conditions can we get the optimal effect of reducing octane loss.

The realization of the optimal operating condition calculated above is conducive to the following aspects:

(1) Improving the quality of refined oil products of chemical enterprises. By debugging the above operating points and controlling the corresponding operating conditions such as temperature, flow, or pressure difference, enterprises can improve the quality of the finished oil.

(2) Saving the economic cost of chemical enterprises. By optimizing operating conditions to reduce the loss of octane number, the economic benefits of enterprises can be improved.

(3) Protecting the atmospheric environment. By optimizing the operating conditions to make the refined oil meet the national standards, it is beneficial to reduce sulfide emissions and reduce air pollution.

7.2. Future Outlook. In the future, research can be carried out in the following directions.

Firstly, combining the catalytic cracking process and principles based on dimensionality reduction can better select the variables of the structural model.

Secondly, the neural network has relatively high requirements for data, but the quality and accuracy of the data in the example are not enough, so the model has a low degree of fit. In the future, more data can be collected or the data preprocessing process can be optimized to obtain data with fewer empty values and outliers.

Thirdly, with the development of data analysis technology, other intelligent machine learning algorithms can be used in the future, and comparative analysis can be performed to select better methods [47, 48].

Finally, for the automobile industry, research can be developed on how to reduce the energy consumption of automobile engines. Reducing energy consumption by improving engine performance is also an important way to reduce gasoline consumption and air pollution.

Appendix

A. Locally Linear Embedding

Locally linear embedding is [36, 37] as follows.

A.1. Locally Linear Range. Firstly, the locally linear range is solved by using the K-nearest neighbor principle. Because of locally linearity, each data point $x_i$can be represented by a linear combination of its nearest neighbor data points. That is, $x_i = \sum_{j=1}^{k} w_{ji} x_{ji}$, and $N_i = knn(x_i, k)$, $N_i = [x_{1i}, \ldots, x_{ki}]$, where w_i is $k \times 1$ column vector, w_{ji} is row j of w_i,

x_{ji} is the j th nearest neighbor to x_i, $(1 \le j \le k)$, $w_i = [w_{1i}, \ldots, w_{ki}]^T$ and, $x_i = [x_{1i}, \ldots, x_{Di}]^T$, where D is the dimension of x_i

Solving the weight coefficient matrix means to solve the constrained optimization problem as follows:

$$\begin{aligned} &\arg\min \sum_{i=1}^{N} x_i - \sum_{j=1}^{k} w_{ji} x_{ji}^2, \\ &\text{s.t.} \quad \sum_{j=1}^{K} w_{ji} = 1. \end{aligned} \tag{A.1}$$

Thus, the expression of the weight coefficient matrix can be deduced as follows:

$$\begin{aligned} \phi(w) &= \sum_{i=1}^{N} x_i - \sum_{j=1}^{k} w_{ji} x_{ji}^2 = \sum_{i=1}^{N} x_i - \sum_{j=1}^{k} w_{ji}\left(x_i - x_{ji}\right)^2, \\ &\sum_{i=1}^{N} X_i - \sum_{j=1}^{k} w_i \left(X_i - N_i\right)^2, \\ X_i &= \left[x_i, \ldots, x_i\right], N_i = \left[x_{1i}, \ldots, x_{ki}\right], \\ &= \sum_{i=1}^{N} w_i^T \left(X_i - N\right)^T \left(X_i - N\right) w_i. \end{aligned} \tag{A.2}$$

Then, view S_i as local covariance matrix, $S_i = (X_i - N_i)^T (X_i - N_i)$. Eq. (A.2) can be viewed as follows:

$$\phi(w) = \sum_{i=1}^{N} w_i^T S_i w_i. \tag{A.3}$$

Use the Lagrange multiplier,

$$L(w_i) = w_i^T S_i w_i + l\left(w_i^T 1_k - 1\right), \tag{A.4}$$

where $1_k$is $k \times 1$ column vector with entries of 1.

Take derivative of Eq. (A.4) as follows:

$$\begin{aligned} \frac{\partial L(w_i)}{\partial w_i} &= 2S_i w_i + l 1_k = 0, \\ w_i &= \frac{S_i^{-1} 1_k}{1_k^T S_i^{-1} 1_k}. \end{aligned} \tag{A.5}$$

A.2. Low-Dimensional Representation. The low-dimensional representation should have the same locally geometric property, so the same linear representation expression is used to finally form the quadratic form. Therefore, it is mapped to a low-dimensional space to solve the following constrained optimization problem:

$$\begin{aligned} &\arg\min_{Y} y(Y) = \sum_{i=1}^{N} y_i - \sum_{j=1}^{k} w_{ji} y_{ji}^2, \\ &\text{s.t.} \quad \sum_{j=1}^{k} y_i y_i^2 = N I_{d \times d}. \end{aligned} \tag{A.6}$$

The output result is the $d \times N$ matrix $Y = [y_1, y_2, \ldots, y_N]$, which is composed of low-dimensional space vectors. Use sparse matrix W to represent w as follows:

$$\begin{aligned} &\sum_{j=1}^{N} w_{ji} y_{ji} = \sum_{j=1}^{k} w_{ji} y_{ji} = Y W_i, \\ &\begin{cases} W_{ji} = w_{ji}, & j \text{ is the nearest neighbor to } i, \\ W_{ji} = 0, & \text{others}, \end{cases} \end{aligned} \tag{A.7}$$

where W_i is the i column of $W\,(N \times N)$, I_i is the i column of unit matrix $I\,(N \times N)$, and y_i is the i column of Y.

So,

$$y(Y) = \sum_{i=1}^{N} Y\left(l_i - W_i\right)^2 = tr\left[Y(I - W)(I - W)^T Y^T\right]. \tag{A.8}$$

Make $M = (I - W)(I - W)^T$ and use the Lagrange multiplier again as follows:

$$L(Y) = YMY^T + l\left(YY^T - NI\right). \tag{A.9}$$

Take derivative of Eq. (A.9) as follows:

$$\frac{\partial L}{\partial Y} = 2MY^T + 2lY^T = 0. \tag{A.10}$$

A.3. Take Eigenvectors. It can be seen that Y is actually the matrix composed of the Eigen vectors of M. To reduce the data to D dimension, we only need to take the minimum d eigenvectors corresponding to nonzero Eigen values of M. Generally, the first minimum Eigen value is close to 0, so we abandon it. Finally, we take the eigenvectors corresponding to the previous $[2, d+1]$ Eigen values from the smallest to the largest.

B. A BP Neural Network Based on Genetic Algorithm

BP neural network based on the genetic algorithm is [38–45] as follows.

B.1. Initialization Network. Firstly, determine the number of network nodes, training set, and test set, and the data are normalized. Then, the individual in the genetic algorithm is initialized with real coding. Each individual is composed of four parts, namely, the weight matrix W_a between the input layer and the hidden layer, the threshold vector q_a of the hidden layer, the weight matrix W_b between the hidden layer and the output layer, and the threshold vector q_b of the output layer, forming a definite neural network.

B.2. BP Neural Network. (1) Forward propagation:

The input layer is $x_1, x_2, \ldots, x_n$, with the weight matrix W_a between the input layer and the hidden layer. Use the linear weighted sum method to obtain the net input of the i neuron of the hidden layer which is $\text{Netin}_i = \sum_{j=1}^{n} x_j W_{a_{ij}}$. The net input is compared with the threshold vector q_a of the hidden layer, and then the neuron output can be obtained by activation function. Sigmod function $f(x) = 1/1 + \exp(-x)$ is selected here as the activation function to realize signal transformation.

Therefore, the output of i neuron of the hidden layer is as follows:

$$y_i = f\left(\text{Netin}_i - q_{a_i}\right). \tag{B.1}$$

Then, with the weight matrix W_b between the hidden layer and the output layer, use the same linear weighted sum method to obtain the net input of the i neuron of the output layer which is $\text{Outnetin}_i = \sum_{j=1}^{n} y_j W_{b_{ij}}$. The net input is compared with the threshold vector q_b of the output layer likewise, and then the i neuron output can be obtained by inverse function of Sigmod as follows:

$$p_i = f^{-1}\left(\text{Outnetin}_i - q_{b_i}\right). \tag{B.2}$$

The real output is $z_1, z_2, \ldots, z_m$, and the least square error of the kth prediction result is as follows:

$$E_K = \frac{1}{2}\sum_{i=1}^{m}\left(p_i - z_i\right). \tag{B.3}$$

(2) Backward propagation:

The purpose of backward propagation is to reduce the prediction error E_K and optimize the neural network. Here, the gradient descent method is used to update and reduce the parameters. The parameter adjustment formula is as follows:

$$P = -h\left(\frac{\partial E_K}{\partial P}\right), \tag{B.4}$$

where h is the leaning rate.

After adjusting the aforementioned parameters, the neural network is optimized by repeating iteration and adjustment.

(3) Termination conditions:

When one of the following termination conditions is reached, the iteration is stopped and a neural network is obtained.

(1) The minimum error threshold $E_{\min}$ is reached
(2) The maximum number of iterations $G_{\max}$ is reached

B.3. Calculate Fitness. After running the neural network with the training set data, the prediction system output was obtained for each individual and the fitness of the individual F_i is defined as the error E_i between the predicted output and the real output, and the formula is as follows:

$$F_i = E_i = \frac{1}{2}\sum_{i=1}^{n}\left(p_i - z_i\right), \tag{B.5}$$

where n is the number of output nodes; p_i is the predicted output of the i node, and z_i is the real output of the i node.

B.4. Selection, Crossover, and Mutation. (1) Selection:

Selection means to retain the high-quality individual, while eliminating the poor quality individual. The Roulette algorithm is chosen for selection operation.

Firstly, calculate the probability that an individual will be selected:

$$p_i = \frac{f_i k}{\sum_{i=1}^{N} f_i}, \tag{B.6}$$

where $f_i = (k/P_i)$ is the reciprocal of the fitness, which is positively correlated with adaptability; k is the coefficient, which is set at 1; and N is the population size.

Then, the interval $[0, 1]$ is divided into N intervals, the length of which is the same as the probability of an individual will be selected. The location interval is determined by generating random numbers so that corresponding individuals could be selected to form a new population.

(2) Cross:

Cross is individuals of new populations exchange gene fragments to generate new individuals. Since individuals are real coding, the real cross method is adopted here for crossover operation.

Firstly, the individual in the group is judged to carry out the cross with the set crossover probability. The formula of cross between the m individual and the n individual at the j intersection position is as follows:

$$\begin{cases} a_{mj} = a_{mj}(1-b) + a_{nj}b, \\ a_{nj} = a_{nj}(1-b) + a_{mj}b, \end{cases} \tag{B.7}$$

where b is the random number in $[0, 1]$.

(3) Mutation:

Mutation is changes in the individual's own gene value. Since individuals are real coding, real value mutation is adopted here for mutation operation.

Firstly, the individual in the group is judged to carry out the mutation with the set mutation probability. Then, the j mutation location of the i individual was selected for the mutation operation. The formula is as follows:

$$a_{ij} = \begin{cases} a_{ij} + \left(a_{ij} - a_{\max}\right) \times f(g), & r > 0.5, \\ a_{ij} + \left(a_{\min} - a_{ij}\right) \times f(g), & r \le 0.5, \end{cases} \tag{B.8}$$

where $a_{\max}$ is the upper bound of a_{ij}, $a_{\min}$ is the lower bound of a_{ij}, r is the random number in $[0, 1]$, $f(g)$ is the mutation formula: $f(g) = r_2(1 - (g/G_{\max}))^2$, r_2 is the random number in $[0, 1]$, g is the current iteration number, and $G_{\max}$ is the maximum number of iterations.

When one of the following termination conditions is reached, the iteration is stopped, and the optimal initial weight and threshold are obtained.

(1) The fitness of the optimal individual and the fitness of the population stopped rising
(2) The maximum number of iterations G_{max} is reached

B.5. Optimal Prediction Model Building. When the optimal initial weight and threshold are obtained by the genetic algorithm above, they are given to BP neural network for the optimal network forming. When one of termination conditions is reached, iteration is stopped, and the optimal neural network is obtained.

Data Availability

Historical accumulated data of a petrochemical company's S-Zorb device are collected, including operating variable data, raw material data, product data, and catalyst data. After adjusting the data frequency difference, 325 samples are formed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Social and Scientific Fund Program of China (17BGL142 and 18ZDA052) and Natural Science Foundation of China (91546117 and 71904117).

Supplementary Materials

(1) S-Table1: the historical accumulated data of the catalytic cracking gasoline S-Zorb unit. There are 325 samples of operating variable data, raw material data, product data, and catalyst data observed in different time intervals within 2017-2020. (2) S-Table2: the certain operating range and minimum adjustment range of operating variables. (*Supplementary Materials*)

References

[1] OICA, *Registrations or Sales of New Vehicles*, Organisation Internationale des Constructeurs d'Automobiles, Paris, France, 2019.
[2] IEA, *Oil Information: Database Documentation*, International Energy Agency, Paris, France, 2020.
[3] R. A. Field, J. N. Lester, S. O. Baek, M. E. Goldstone, P. W. Kirk, and R. Perry, "A review of atmospheric polycyclic aromatic hydrocarbons: sources, fate and behavior," *Water Air & Soil Pollution*, vol. 60, no. 3-4, pp. 279–300, 1991.
[4] Health Effects Institute, "Traffic-related air pollution: a critical review of the literature on emissions, exposure, and health effects," *Environment*, vol. 131, pp. 409–444, 2010.
[5] N. R. Khalili, P. A. Scheff, and T. M. Holsen, "PAH source fingerprints for coke ovens, diesel and, gasoline engines, highway tunnels, and wood combustion emissions," *Atmospheric Environment*, vol. 29, no. 4, pp. 533–542, 1995.
[6] R. K. Larsen and J. E. Baker, "Source apportionment of polycyclic aromatic hydrocarbons in the urban atmosphere: a comparison of three methods," *Environmental Science & Technology*, vol. 37, no. 9, pp. 1873–1881, 2003.
[7] M. Riedl and D. Diaz-Sanchez, "Biology of diesel exhaust effects on respiratory function," *Journal of Allergy and Clinical Immunology*, vol. 115, no. 2, pp. 221–228, 2005.
[8] IEA, *World Energy Balances: Database Documentation*, International Energy Agency, Paris, France, 2020.
[9] F. Ouyang, W. Fang, J. Tang, and H. Jiang, "Optimizing product distribution of mip process using BP neural network," *Petroleum Processing and Petrochemicals*, vol. 47, no. 5, pp. 95–100, 2016.
[10] E. Paranghooshi, M. T. Sadeghi, and S. Shafiei, *Prediction of Octane Number for Gasoline Blends Using Artificial Neural Networks and Genetic Algorithms*, ACS Publications, Washington, DC, USA, 2009.
[11] X. J. Qin and Z. H. Chen, "The design of the neural network model for predicting gasoline RON," *Controland Decision*, vol. 14, no. 2, pp. 151–155, 1999.
[12] J. Tian, X. Du, Y. Zheng, Y. Li, and T. Jing, "Prediction of sulfur content in hydrodesulfurization diesel oil based on the PSO-BP neural network," *Petrochemical Technology*, vol. 46, no. 1, pp. 62–67, 2017.
[13] Y. L. Yang, X. D. Zhang, and H. Jiang, "BP network based on GA-BP optimization algorithm and application in gasoline blending octane number modeling," *Microcomputer Information*, vol. 22, no. 11, pp. 276–278, 2006.
[14] S. Brunet, D. Mey, G. Pérot, C. Bouchy, and F. Diehl, "On the hydrodesulfurization of FCC gasoline: a review," *Applied Catalysis A: General*, vol. 278, no. 2, pp. 143–172, 2005.
[15] J. A. Valla, A. A. Lappas, I. A. Vasalos, C. W. Kuehler, and N. J. Gudde, "Feed and process effects on the in situ reduction of sulfur in FCC gasoline," *Applied Catalysis A General*, vol. 276, no. 1-2, pp. 75–87, 2004.
[16] Y. N. Yang, Y. Ren, A. G. Mao, and H. P. Tian, "Analysis of technical factors affecting RON of FCC gasoline," *Petroleum Refinery Engineering*, vol. 1, no. 6, pp. 32–35, 2019.
[17] S. M. Jacob, B. Gross, S. E. Voltz, and V. W. Weekman, "A lumping and reaction scheme for catalytic cracking," *AIChE Journal*, vol. 22, no. 4, pp. 701–713, 1976.
[18] L. G. Lin, G. Wang, H. M. Qu et al., "Pervaporation performance of cross-linked polyethylene glycol membranes for deep desulfurization of FCC gasoline," *Journal of Membrane Science*, vol. 280, no. 1-2, pp. 651–658, 2006.
[19] C. Song, "An overview of new approaches to deep desulfurization for ultra-clean gasoline, diesel fuel and jet fuel," *Catalysis Today*, vol. 86, no. 1-4, pp. 211–263, 2003.
[20] M. A. B. Siddiqui and A. M. Aitani, "FCC gasoline sulfur reduction by additives: a review," *Petroleum Science and Technology*, vol. 25, no. 3, pp. 299–313, 2007.
[21] J. A. Salazar, L. M. Cabrera, E. Palmisano, W. J. Garcia, and R. B. Solari, "Process for producing reformulated gasoline by reducing sulfur, nitrogen and olefin," U. S. Patent 5770047, 1998.

[22] M. F. Li, G. F. Xia, Y. Chu, and Y. J. Hu, "Development of RSDS-Icatalyst for selective hydrodesulfurization of FCC gasoline," *Petroleum Processing and Petrochemicals*, vol. 34, no. 7, pp. 1–4, 2003.

[23] H. Li, W. Shan, S. Shen et al., "Production of a gasoline blending component with high-octane and low sulfur from coal tar light oil over sulfided CoMoP/η-Al2O3," *Journal of Cleaner Production*, vol. 228, no. 10, pp. 965–973, 2019.

[24] S. Ayoub and V. Masoud, "Design and optimization of hydrodesulfurization process for liquefied petroleum gases," *Journal of Cleaner Production*, vol. 220, no. 20, pp. 1255–1264, 2019.

[25] N. Hasheminejad, H. Tavakol, and W. Salvenmoser, "Preparation of gold-decorated simple and sulfur-doped carbon spheres for desulfurization of fuel," *Journal of Cleaner Production*, vol. 264, no. 10, Article ID 121684, 2020.

[26] X. Qin, J. Liu, C. Wang et al., "Molecular level analysis on performance of diameter expanding reactor to improve gasoline quality in FCC process," *Fuel*, vol. 290, Article ID 119978, 2021.

[27] M. G. Li, Y. Y. Xu, J. Men et al., "Hybrid variable selection strategy coupled with random forest (RF) for quantitative analysis of methanol in methanol-gasoline via raman spectroscopy," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 251, no. 1, Article ID 119430, 2021.

[28] G. F. Wei, "Study on gas recognition method with principal component analysis and back-propagation neural network," *Journal of Transcluction Technology*, vol. 12, no. 04, pp. 292–298, 2001.

[29] H. X. Cheng and F. Yi, "Light gasoline etherification predictive control with BP neural network model," *Automation in Petro-Chemical Industry*, vol. 48, no. 6, pp. 40–42, 2012.

[30] Z. Zhang, Z. Li, Y. Li, and G. Li, "GA-ANN method for prediction of gasoline yield of RFCCU," *Petroleum Processing and Petrochemicals*, vol. 45, no. 7, pp. 91–96, 2014.

[31] X. Su, H. J. Pei, Y. Y. Wu, J. S. Gao, and X. Y. Lan, "Predicting coke yield of FCC unit using genetic algorithm optimized BP neural network," *Chemical Industry and Engineering Progress*, vol. 20, no. 2, pp. 389–396, 2016.

[32] H. Cheng, Y. Zhang, L. Kong, and X. Y. Meng, "The application of neural network PID controller to control the light gasoline etherification," *Iop Conference Series Earth and Environmental Science*, vol. 69, no. 1, Article ID 012045, 2017.

[33] Y. Zhang, L. Zhao, F. Chen, Y. Wang, and C. Xu, "High efficiency separation of olefin from FCC naphtha: separation mechanism and universal simulation method," *AIChE Journal*, vol. 67, no. 5, Article ID e17153, 2021.

[34] B. Foroughi, J. R. Shahrouzi, and R. Nemati, "Detection of gasoline adulteration using modified distillation curves and artificial neural network," *Chemical Engineering & Technology*, vol. 44, no. 3, pp. 527–534, 2021.

[35] Y. Ma, Z. Yu, Y. Wang, D. Xie, and J. E, "Investigation on the influence of initial thermodynamic conditions and fuel compositions on gasoline octane number based on a data-driven approach," *Fuel*, vol. 291, Article ID 120124, 2021.

[36] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[37] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," *Journal of Machine Learning Research*, vol. 7, pp. 1–13, 2001.

[38] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[39] J. Savković -Stevanović, "Neural networks for process analysis and optimization: modeling and applications," *Computers & Chemical Engineering*, vol. 18, no. 11-12, pp. 1149–1155, 1994.

[40] C. H. Chen, "An explainable deep neural network for extracting features," *Science eLetters*, vol. 365, no. 9, p. 6452, 2019.

[41] C. Guo, G. Liu, L. Lyu, and C.-H. Chen, "An unsupervised PM2.5 estimation method with different spatio-temporal resolutions based on KIDW-TCGRU," *IEEE Access*, vol. 8, pp. 190263–190276, 2020.

[42] D. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley, Boston, MA, USA, 1989.

[43] W. Gao, "An improved fast-convergent genetic algorithm and its performance study," *Systems Engineering and Electronics*, vol. 25, no. 11, pp. 1427–1430, 2003.

[44] R. Chen, Y. Xu, and H. Lan, "Research on multilayer feed forward neural networks-genetic backpropagation algorithm & structure optimization strategy," *Acta Automatica Sinica*, vol. 23, no. 1, pp. 43–49, 1997.

[45] C.-H. Chen, F. Song, F.-J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, no. 541, Article ID 123344, 2020.

[46] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[47] X. Wu, Y. R. Cao, Y. Xiao, and J. Guo, "Finding of urban rainstorm and waterlogging disasters based on microblogging data and the location-routing problem model of urban emergency logistics," *Annals of Operations Research*, vol. 60, pp. 1–32, 2018.

[48] X. Wu, Z. Xu, H. Liu, J. Guo, and L. Zhou, "What are the impacts of tropical cyclones on employment? an analysis based on meta-regression," *Weather, Climate, and Society*, vol. 11, pp. 259–275, 2019.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 4439861, 9 pages
https://doi.org/10.1155/2021/4439861

Research Article

Matching Transportation Ontologies with Word2Vec and Alignment Extraction Algorithm

Xingsi Xue,[1,2] **Haolin Wang**,[1,2] **Jie Zhang**,[3] **Yikun Huang**,[4] **Mengting Li**,[2] **and Hai Zhu**[5]

[1]*Intelligent Information Processing Research Center, Fujian University of Technology, Fuzhou, Fujian 350118, China*
[2]*School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou, Fujian 350118, China*
[3]*School of Computer Science and Engineering, Yulin Normal University, Yulin, Guanxi 537000, China*
[4]*Concord University College Fujian Normal University, Fuzhou, Fujian 350117, China*
[5]*School of Network Engineering, Zhoukou Normal University, Zhoukou, Henan 466001, China*

Correspondence should be addressed to Yikun Huang; fjnuhyk@163.com

Received 10 April 2021; Accepted 4 May 2021; Published 11 May 2021

Academic Editor: Chi-Hua Chen

The development of intelligent transportation systems (ITSs) faces the challenge of integrating data from multiple unrelated sources. As one of the core technologies of knowledge integration in ITS, an ontology typically provides a normative definition of transportation domain that can be used as a reference for information integration. However, due to the subjectivity of domain experts, a concept may be expressed in multiple ways, yielding the ontology heterogeneity problem. Ontology matching (OM) is an effective method of addressing it, which is of help to further realize the mutual communication between the ontology-based ITSs. In this work, we first propose to use Word2Vec to model the entities in vector space and calculate their similarity values. Then, a stable marriage-based alignment extraction algorithm is presented to determine high-quality alignment. In the experiment, the performance of the proposal is tested by using the benchmark track of OAEI and real transportation ontologies. The experimental results show that our approach is able to obtain higher quality alignment results than OAEI's participants and other state-of-the-art ontology matching techniques.

1. Introduction

Data in the transportation domain are complex and varied [1–3]. These data come from a variety of data collection methods such as traffic sensors, surveys, and devices [4]. Therefore, the development of intelligent transportation systems (ITSs) faces the challenge of integrating data from multiple unrelated sources [5–7]. These data are semantically imprecise, conceptually ambiguous, and informative. As one of the core technologies of knowledge integration in ITS, an ontology typically provides a formal and normative definition of domain knowledge [8–10]. They enable collaboration between ontology-based ITSs by defining related concepts in the domain and the relationships between concepts [4]. However, due to the subjectivity of domain experts, the notion of a concept might be expressed in multiple ways [11, 12]. In order to achieve mutual communication between ontology-based ITSs, it is important to determine the logical relationships between heterogeneous ontologies [13]. Ontology matching (OM) is an important technique to solve the problem of semantic heterogeneity [14, 15], which is dedicated to discovering correspondences between related entities (e.g., classes and properties) in different ontologies [16]. For this reason, it is effective to use ontology matching techniques to solve the existing semantic heterogeneity of transport ontologies.

In recent years, researchers have proposed a large number of ontology alignment strategies and developed various semiautomated or automated ontology matching systems [17–19]. However, the existing ontology matching schemes have many drawbacks: the matcher's poor ontology similarity calculation, inefficient extraction of ontology mapping results, etc. To address these drawbacks, we first propose to use Word2Vec [20] to model the entities in vector

space, calculate their similarity values, and use Wikipedia training data to improve the model's generalizability and the alignment's quality. Moreover, a stable marriage-based ontology extraction algorithm is presented to improve the quality of alignment.

The rest content of this article is as follows. In Section 2, we briefly describe the application of the ontology matching to transportation data. Section 3 presents ontology and ontology matching in detail. In Section 4, we develop an ontology matching system using the Word2Vec model. We propose Word2Vec-based similarity measure and stable marriage-based ontology extraction algorithm in Sections 5 and 6. In Section 7, experimental results and analysis are described. Finally, conclusion and future work are provided in Section 8.

2. Related Work

Ontology matching is well suited to solve the problems arising from semantic ambiguity and large data volume in transportation data [21–23]. Benvenuti et al. [24] integrated Transmodel ontologies and KPIOnto to facilitate the study of public road monitoring systems. The Transmodel is a reference data model about the European public transport information system that represents traffic ontologies and their relationships. Bermejo et al. [25], in order to avoid the use of a central decision point in a traffic network, proposed to treat each vehicle as an ontology that gives it reasoning capabilities. In emergency traffic control, each vehicle in the proposed system is a decision point that considers the state and location of neighboring vehicles and collaborates with them to reach consensus in real time. Overall, ontology provides a prescriptive approach to the development of knowledge in the transportation domain that can support integrated information in a variety of ways. Standardization efforts in transportation can be greatly assisted by the ontology engineering approach.

Recently, the technique for calculating words' similarity and relevance using the Word2Vec model from word embedding has become a research hotspot and is gradually applied to the domain of ontology matching. Xue and Pan [26] modeled the ontology in vector space and then used the linguistic information of the entities to reduce the dimensionality, which improved the efficiency of similarity calculation and entity matching. Zhang et al. [27] introduced word embedding techniques to the field of ontology matching and proposed a hybrid method that incorporates word embedding into the calculation of semantic similarity between elements. Teslya and Savosin [28] proposed a Word2Vec vector-based language model for the ontology mapping problem. The model extends on the basis of specific ontology relations. The semantics of the language is used to match ontologies without considering the form of words or specific terms. It can be seen that it is feasible to calculate the semantic similarity using the Word2Vec model. As for alignment result extraction, with a large number of matching methods being proposed, researchers usually need to integrate multiple strategies to improve the quality of alignment. To better extract matching results, a stable marriage-based ontology extraction algorithm is proposed in this work, which further improves the performance of the matcher.

3. Ontology and Ontology Matching

An ontology is a conceptualized normative description on the domain knowledge [29–31]. Specifically, an ontology normatively defines classes, properties, other entities in a domain [32, 33], and the relationships between them. Figure 1 shows the ontology in a road accident [34]. The words in the rectangular box are classes, e.g., "Ting," "Vehicle," and "Insurance Company." The hollow arrows indicate the structural relationship between two classes, e.g., "Official Agency" is a subclass of "Insurance Company" and "Road Accident" is a subclass of "Event." The black arrows represent properties that describe the relationship between the two classes. However, the same entity could be constructed in multiple ways in different ontologies, yielding the problem of semantic heterogeneity between ontologies.

In order to illustrate the matching problem, the result of matching between two simple ontologies O and O' is presented in Figure 2. The two ontologies in the figure have descriptions of classes, properties, and instances. Classes are displayed in rectangles. Structure-based relationships are shown as broken line arrows. In O, "Chairman" is a specialization (subclass) of "Person." Correspondences are shown as blue double arrows that connect classes from O to classes from O' and depict their relationships. There are symbols: $\perp$, $\subseteq$ (or $\supseteq$), and $\equiv$, which mean disjointness, more specific (or less specific), and equivalence relation, respectively. For example, the "Subject Area" in one ontology is equivalent to the "Topic" in another ontology, and the "Regular Author" is an irrelevant relationship with the "Reviewer."

4. The Framework of W2V-OM

This work constructs an ontology matcher (W2V-OM) that calculates the similarity values of two entities using the Word2Vec model, as shown in Figure 3. The dataset for training the Word2Vec model is the English Language Wikipedia articles in the Wikipedia database [35]. The corpus is universal and can cope with language processing problems in many domains. These textual data are unstructured and need to be preprocessed into structured data. After the model is trained, the source transportation ontology and the target ontology are parsed. The entities extracted from the ontology are fed into the word2vec model to calculate the cosine similarity and integrate the linguistic-based similarity measure to yield the similarity matrix. Then, the ontology mapping results are obtained using the stable marriage-based ontology extraction algorithm. Finally, the ontology matching quality is evaluated based on reference alignment.

5. Word2Vec-Based Similarity Measure

The similarity measure is a function where the information of two ontology entities is used as input and a real number between [0, 1] is output to represent their similarity [36]. Specifically, the closer the result is to 1, the more similar they

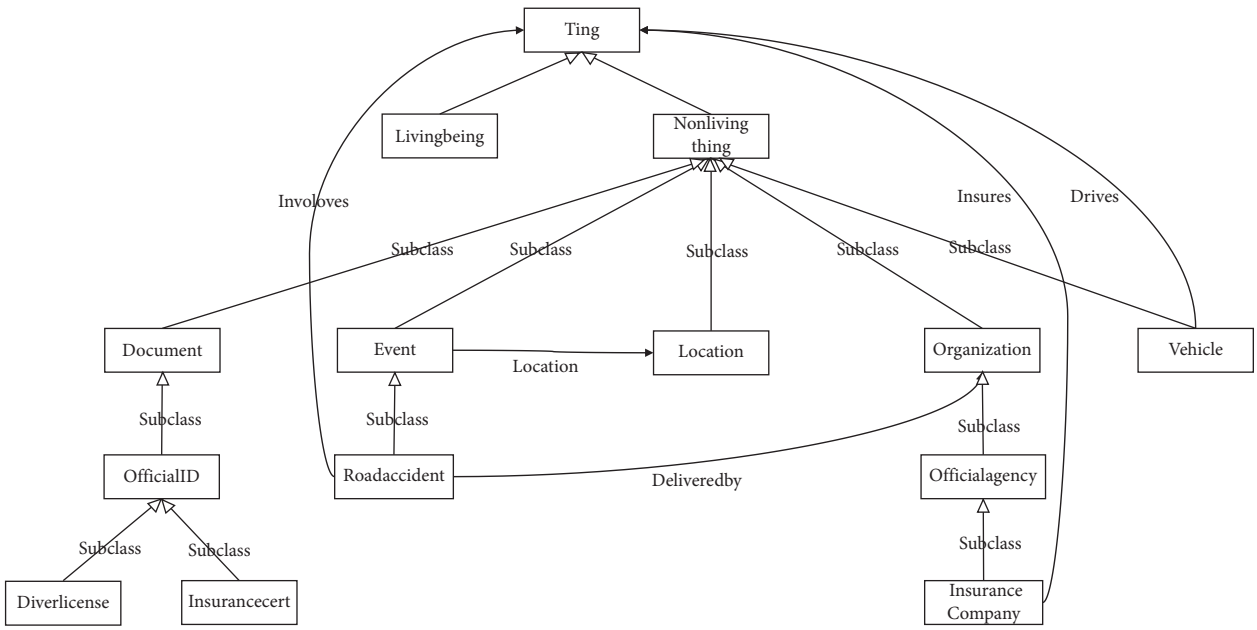


Figure 1: An example of a road accident ontology.

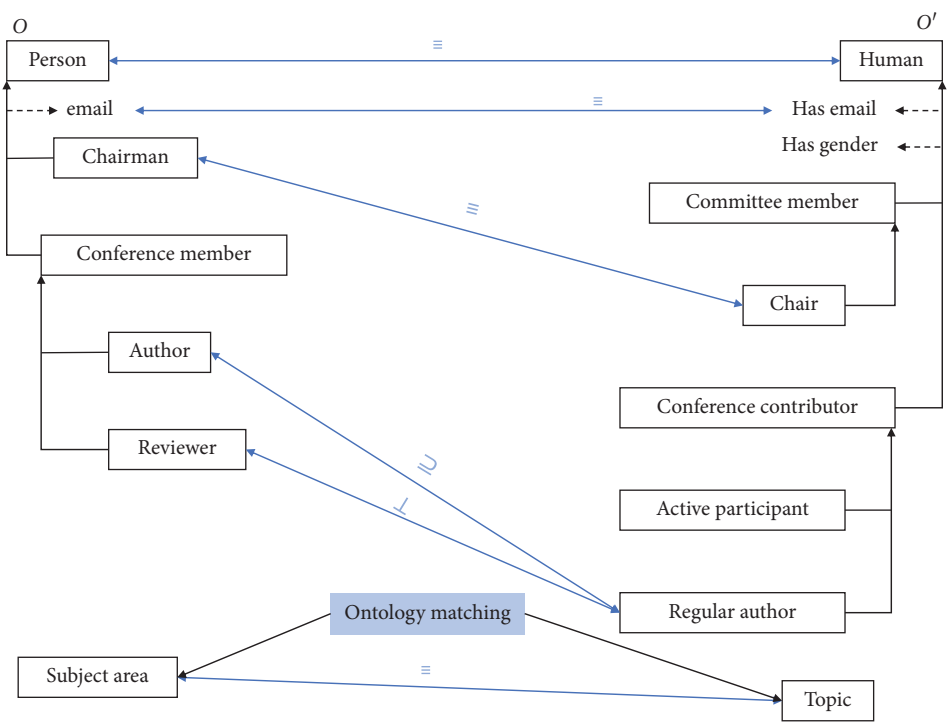


Figure 2: An example of ontology alignment.

are; the closer the result is to 0, the less similar they are. Similarity measure is an important part of the ontology matching process. Utilizing different similarity measure affects the results of ontology alignment. In this work, we use two categories of similarity measures to calculate the similarity values of two entities, i.e., linguistic-based measure and cosine similarity measure using the Word2Vec model.

Word2Vec is a language model Natural Language Processing (NLP) where words or phrases are represented as real number vectors. Similar words usually have the proximity of vectors and are mapped to the same region, as shown in Figure 4. With regard to the ontology representation in vector space, it means that a class or a property of ontologies can be represented in dimensions of the vector space. Specifically, the different classes or properties are uniquely represented in the vector space. The vector space covers all classes and properties in both ontologies. In this work, the dimensions of the vector space are determined by all the classes and properties in the

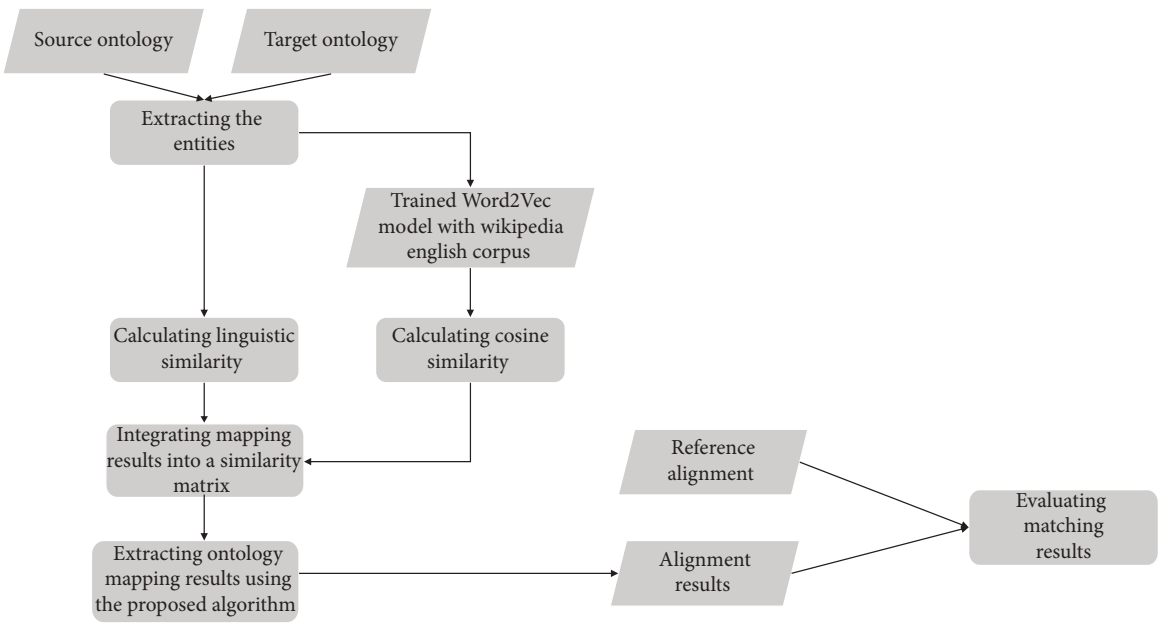


Figure 3: The framework of W2V-OM.

two ontologies. The Word2Vec model is trained using the Wikipedia English corpus. Each entity is represented as a vector in vector space, and then, the similarity of the two entities is calculated using the cosine similarity formula. The formula is defined as follows:

$$\text{Cosine Similarity}\left(V_{w_1}, V_{w_2}\right) = \frac{V_{w_1} \cdot V_{w_2}}{V_{w_1} \cdot V_{w_2}}, \quad (1)$$

where V_{w_1} and V_{w_2} are, respectively, the vectors of two words w_1 and w_2 and V_{w_1} and V_{w_2}, respectively, denote their norms.

The linguistic similarity between two words is calculated by semantic relations (synonymy and antonymy), which is generally done using dictionaries and lists of synonyms. WordNet [37], a vocabulary database that builds semantic networks based on the semantic information of words, is used to calculate similarity. The linguistic similarity of two words w_1 and w_2 is 1 when w_1 and w_2 are synonyms in WordNet; the similarity is 0.5 when w_1 and w_2 are hypernym in WordNet; in other cases, the similarity is 0.

The two similarity measures produce two similarity matrices, and it is necessary to use an aggregation strategy to set the different matrices into one matrix. In this work, we empirically use the maximum strategy to integrate the similarity measures, i.e., the larger one of two similarity values is selected as the final similarity value, which is of help to ensure the completeness of the alignment.

6. Stable Marriage-Based Alignment Extraction

Integrating the results computed from the similarity measures in a similarity matrix. The ith row and jth column of this matrix represent the entities e_{Si} and e_{Tj} in the source ontology O_S and the target ontology O_T, respectively. The values in the matrix indicate the similarity of the two entities. Larger similarity values indicate higher confidence in the equivalence of the two entities and vice versa indicates less confidence. In this paper, we propose a stable marriage-based ontology extraction algorithm that incorporates a thresholding strategy to obtain better mapping results. The specific steps are as follows: (1) all similarity values in the matrix are sorted in descending order, (2) record the position (e_{Si}, e_{Tj}, m_{ij}) of the maximum similarity in the matrix, where m_{ij} is the maximum similarity, (3) set the value in the same row and column of m_{ij} as 0, and (4) repeat the above three steps until all similarity values in the matrix are 0.

Figure 5 presents the results of extracting the ontology mapping using the proposed method. As shown in the figure, six entity correspondences were finally extracted, which are ($e_{S,1}$, $e_{T,1}$, 0.95), ($e_{S,2}$, $e_{T,2}$, 0.88), ($e_{S,3}$, $e_{T,3}$, 0.6), ($e_{S,5}$, $e_{T,5}$,0.6), ($e_{S,6}$, $e_{T,5}$, 0.6), and ($e_{S,4}$, $e_{T,4}$, 0.1). The proposed algorithm terminates when all similarity values in the matrix are zero, which may result in extracting some entity correspondences with low similarity. For the mapping results, these low similarities are noise. This work therefore incorporates a threshold strategy. A threshold parameter is set and the algorithm is terminated when all values in the similarity matrix are less than the threshold. Assume a threshold of 0.5, i.e., a similarity of less than 0.5 is not reliable. Then, the similarity matrix extraction results are ($e_{S,1}$, $e_{T,1}$, 0.95), ($e_{S,2}$, $e_{T,2}$, 0.88), ($e_{S,3}$, $e_{T,3}$, 0.6), ($e_{S,5}$, $e_{T,5}$, 0.6), and ($e_{S,6}$, $e_{T,5}$, 0.6).

7. Experiment

7.1. Experimental Configuration. In the experiment, the performance of our proposal is tested using real sensor ontologies in the transportation field, as well as the benchmark track provided by the Ontology Alignment Evaluation Initiative (OAEI). The benchmark test library is constructed from reference ontologies in different domains. Each test case in the benchmark track contains two

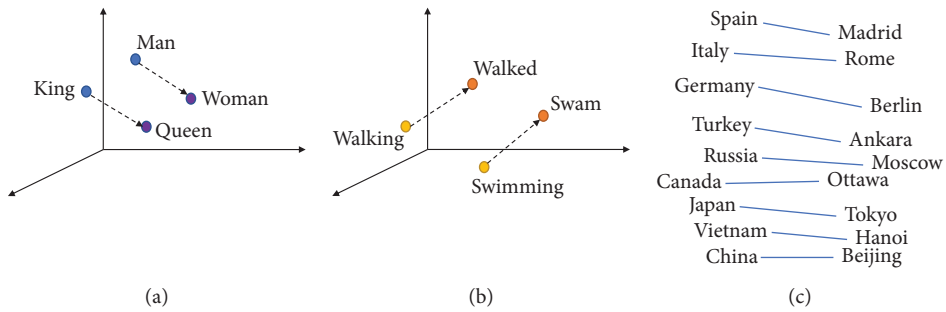


Figure 4: Model the words in vector space. (a) Male-Female. (b) Verb tense. (c) Country-Capital.

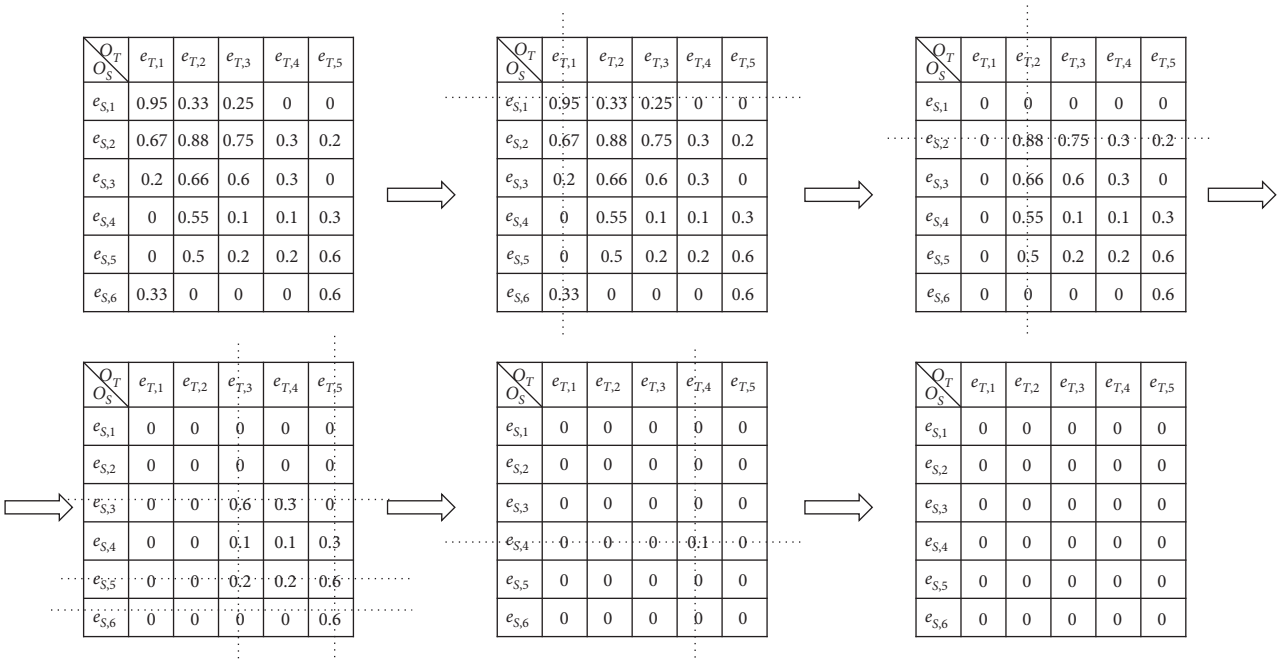


Figure 5: An example of alignment extraction.

ontologies to be matched (a target ontology and a source ontology) and a reference alignment for evaluating the effectiveness of the ontology matcher. The real sensor ontologies used are OSSN, SN, SOSA, and SSN. Table 1 presents a detailed description of the benchmark test cases, and the concise introduction of the sensor ontologies is given in Table 2. In order to assess the quality of ontology matching results, the following are the traditional definitions of ontology alignment metrics:

$$recall = \frac{correct\_found\_correspondences}{all\_possible\_correspondences},$$

$$precision = \frac{correct\_found\_correspondences}{all\_found\_correspondences}, \quad (2)$$

$$f - measure = \frac{2 * reacll * precision}{recall + precesion},$$

where *precision* and *recall* denote the accuracy and completeness of the alignment results, respectively, and $f - measure$ is the harmonic mean of *precision* and *recall* to balance them.

7.2. Comparison with OAEI's Participants. Figures 6–8 present the comparison between W2V-OM and the participants of OAEI in terms of recall, precision, and f-measure, respectively. In the figures, the horizontal axis indicates the testing case ID, the vertical axis indicates the alignment's evaluation metric, and the legends indicate the different matching systems. As shown in the figure, the W2V-OM is higher than the other OAEI's participants in terms of recall and F-measure. With respect to precision, our approach outperforms the other participants in most cases. In summary, the performance of W2V-OM proposed in this work is better than OAEI's participants and can determine high-quality ontology alignment.

Table 1: A detailed description of the benchmark test case.

Cases ID	Case introduction
101–104	Two same ontologies
201–210	Two ontologies with different linguistic and lexical features
221–247	Two different structures of ontologies
248–262	Two ontologies with different structure, linguistic, and lexical features

Table 2: A concise introduction of the sensor ontologies.

Ontology name	Scale	Website
Semantic Sensor Network (SSN) ontology	55 entities	https://www.w3.org/ns/ssn/
Original Semantic Sensor Network (OSSN) ontology	107 entities	https://www.w3.org/2005/Incubator/ssn/wiki/SSN#Sensor
SensorOntology2009 (SN) ontology	152 entities	https://www.w3.org/2005/Incubator/ssn/wiki/SensorOntology2009
Sensor, Observation, Sample, and Actuator (SOSA) ontology	42 entities	https://www.w3.org/ns/sosa/

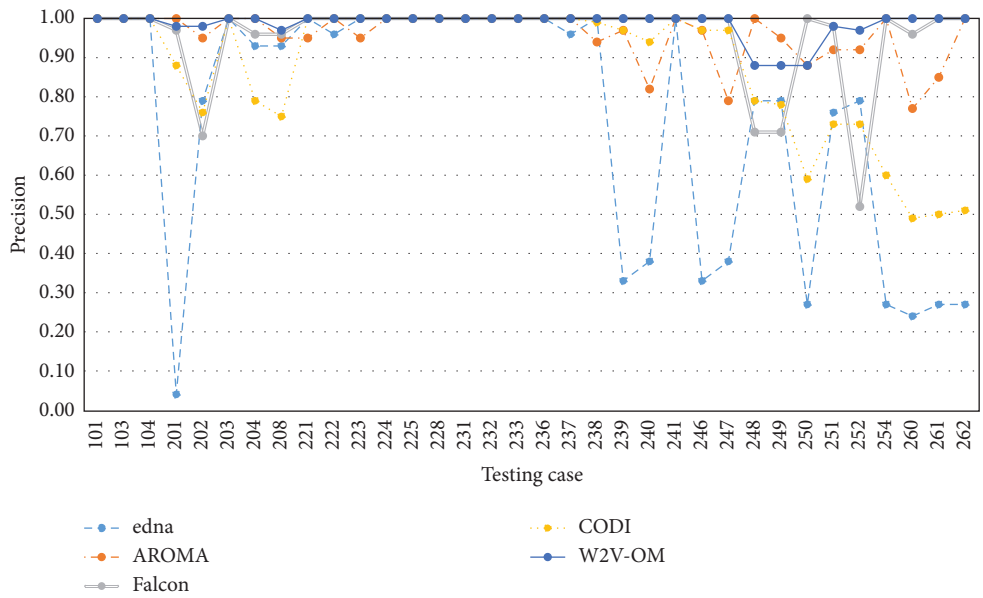


Figure 6: Comparison among W2V-OM and OAEI's participants in terms of precision.

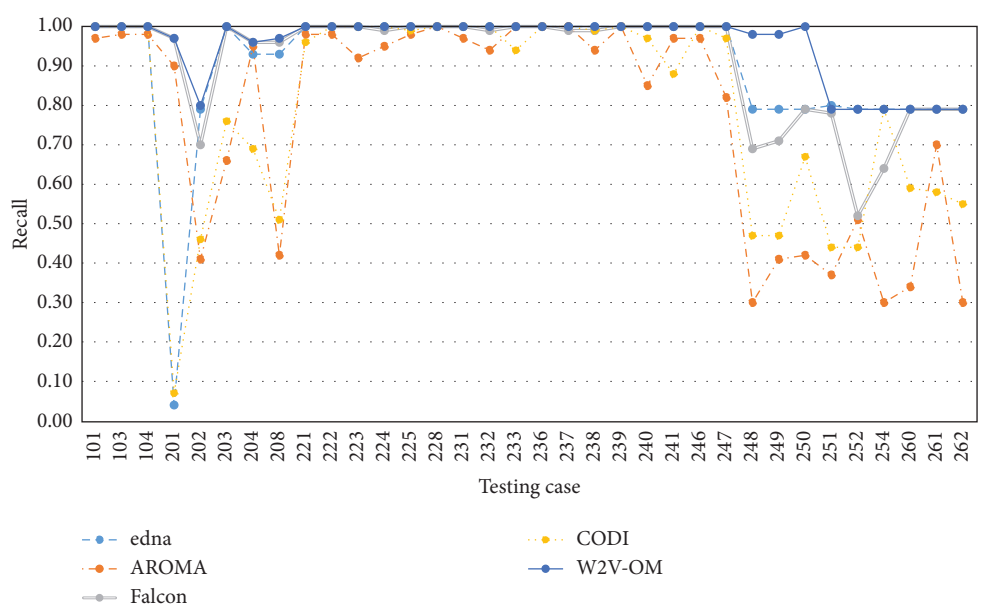


Figure 7: Comparison among W2V-OM and OAEI's participants in terms of recall.

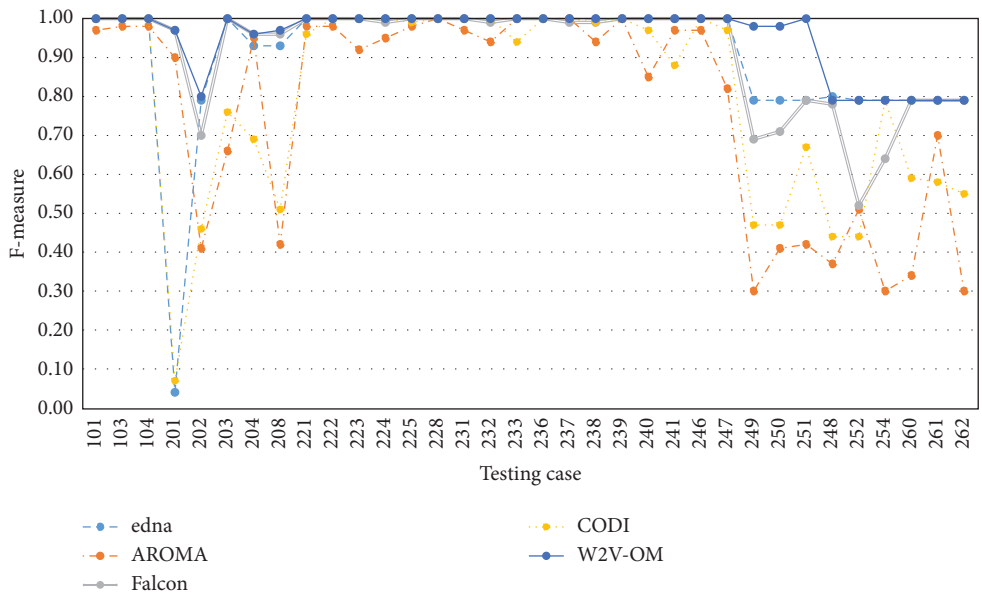


Figure 8: Comparison among W2V-OM and OAEI's participants in terms of F-measure.

Table 3: Comparison among W2V-OM and the state-of-the-art ontology matchers.

Alignment task	Ontology alignment metrics	WordNet-based matcher	SF-based matcher	Jaro-Winkler-based matcher	Levenshtein-based matcher	W2V-OM
SOSA-SN	Recall	1.00	0.07	1.00	1.00	1.00
	Precision	0.33	0.13	0.75	0.75	1.00
	f-measure	0.50	1.00	0.86	0.86	1.00
SOSA-OSSN	Recall	1.00	0.50	1.00	1.00	1.00
	Precision	0.67	0.20	1.00	1.00	1.00
	f-measure	0.80	0.29	1.00	1.00	1.00
SSN-OSSN	Recall	0.97	0.35	1.00	1.00	1.00
	Precision	0.80	0.06	0.94	1.00	1.00
	f-measure	0.88	0.11	0.97	1.00	1.00
SSN-SN	Recall	1.00	0.56	1.00	1.00	1.00
	Precision	0.52	0.02	0.90	1.00	1.00
	f-measure	0.70	0.04	0.95	1.00	1.00

7.3. Comparison with State-of-the-Art Ontology Matchers. Regarding the alignment of sensor ontologies, four popular ontology matchers were used as comparison groups, which are based on WordNet similarity [37], similarity flooding (SF) [38], Jaro–Winkler distance [39], and Levenshtein distance [40]. Table 3 shows the experimental results of sensor ontologies alignment. As can be seen from experimental results, W2V-OM outperforms other methods in four real sensor ontology matching tasks, which demonstrates the effectiveness of our approach.

Since our approach uses Word2Vec to map ontologies into vector space, the similarity is derived by calculating the vector cosine angle of the two entities. The model fully considers the string-based similarity measure and obtains a high similarity. In the mapping result extraction process, the similarity value of the matched entities should be the largest in the same row and column, which means that these two entities are the best alignment. The performance of the matcher is further improved by retaining the entity correspondences with larger similarity values from the similarity matrix using the stable marriage strategy. To sum up, comparison with other matchers demonstrates the effectiveness of the proposed method.

8. Conclusion and Future Work

The purpose of matching transportation ontologies is to determine all the heterogeneous entity pairs. To this end, this work first models entities in vector space with Word2vec and uses the cosine similarity measure to calculate two entities' similarity value. After that, a stable marriage-based alignment extraction algorithm is used to determine high-quality alignment. The experimental results indicate that our approach can obtain higher quality alignment results compared to state-of-the-art ontology matchers and OAEI's participants.

In the future, we will adopt more advanced similarity measures to improve ontology similarity results. We also want to extend ontologies in the transportation domain,

such as the road traffic management ontology and the road accident ontology. Since transportation ontology matching requires particular alignment and knowledge background, specific techniques and strategies need to be proposed to enhance the quality of matching.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province (no. 2020J01875), National Natural Science Foundation of China (nos. 61801527 and 61103143), Fujian Province 13th Five-Year Plan Teaching Reform Project in 2019 (no. FBJG20190156), The Third Batch of Key Lifelong Education Projects in Fujian Province (no. ZS20033), 2018 Program for Outstanding Young Scientific Researcher in Fujian Province University the Research Innovation Team of Concord University College Fujian Normal University in 2020 (no. 2020-TD-001), and Scientific Research Project of Concord University College of Fujian Normal University in 2020 (no. KY20200203).

References

[1] Y. Liu, X. Weng, J. Wan, X. Yue, H. Song, and A. V. Vasilakos, "Exploring data validity in transportation systems for smart cities," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 26–33, 2017.

[2] F. Ali, D. Kwak, P. Khan et al., "Transportation sentiment analysis using word embedding and ontology-based topic modeling," *Knowledge-Based Systems*, vol. 174, pp. 27–42, 2019.

[3] L. Zhu, F. R. Yu, Y. Wang et al., "Big data analytics in intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[4] M. Katsumi and M. Fox, "Ontologies for transportation research: a survey," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 53–82, 2018.

[5] C.-H. Chen, F.-J. Hwang, and H.-Y. Kung, "Travel time prediction system based on data clustering for waste collection vehicles," *IEICE Transactions on Information and Systems*, vol. E102.D, no. 7, pp. 1374–1383, 2019.

[6] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[7] C. H. Chen, F. Song, F. J. Hwang, and L. Wu, "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, pp. 1–10, 2020.

[8] C.-H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 265–267, 2020.

[9] C. Jiang and X. Xue, "A uniform compact genetic algorithm for matching bibliographic ontologies," *Applied Intelligence*, pp. 1–16, 2021.

[10] X. Xue and Y. Wang, "Ontology alignment based on instance using NSGA-II," *Journal of Information Science*, vol. 41, no. 1, pp. 58–70, 2015.

[11] J. S. Pan, P. C. Song, S. C. Chu, and Y. J. Peng, "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, pp. 1–19, 2020.

[12] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, pp. 1–11, 2021.

[13] H. Liu, Y. Wang, and N. Fan, "A hybrid deep grouping algorithm for large scale global optimization," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 6, pp. 1112–1124, 2020.

[14] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, "Integrating sensor ontologies with global and local alignment extractions," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6625184, 10 pages, 2021.

[15] X. Xue, "A compact firefly algorithm for matching biomedical ontologies," *Knowledge and Information Systems*, vol. 62, pp. 1–17, 2020.

[16] X. Xue, C. Yang, C. Jiang et al., "Optimizing ontology alignment through linkage learning on entity correspondences," *Complexity*, vol. 2021, Article ID 5574732, 12 pages, 2021.

[17] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2015.

[18] X. Xue and Y. Wang, "Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[19] X. Xue, H. Yang, J. Zhang, J. Zhang, and D. Chen, "An automatic biomedical ontology meta-matching technique," *Journal of Network Intelligence*, vol. 4, no. 3, pp. 109–113, 2019.

[20] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[21] X. Xue, J. Lu, and J. Chen, "Using NSGA-III for optimising biomedical ontology alignment," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 3, pp. 135–141, 2019.

[22] S. Fan, Z. Hua, V. C. Storey, and J. L. Zhao, "A process ontology based approach to easing semantic ambiguity in business process modeling," *Data & Knowledge Engineering*, vol. 102, pp. 57–77, 2016.

[23] J. C.-W. Lin, Y. Shao, Y. Djenouri, and U. Yun, "ASRNN: a recurrent neural network with an attention model for sequence labeling," *Knowledge-Based Systems*, vol. 212, p. 106548, 2021.

[24] F. Benvenuti, C. Diamantini, D. Potena, and E. Storti, "An ontology-based framework to support performance monitoring in public transport systems," *Transportation Research Part C: Emerging Technologies*, vol. 81, pp. 188–208, 2017.

[25] A. J. Bermejo, J. Villadangos, J. J. Astrain, and A. Córdoba, "Ontology based road traffic management," *Intelligent Distributed Computing VI*, vol. 446, pp. 103–108, 2013.

[26] X. Xue and J. S. Pan, "An overview on evolutionary algorithm based ontology matching," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 1, pp. 75–88, 2018.

[27] Y. Zhang, X. Wang, S. Lai et al., "Ontology matching with word embeddings," *Lecture Notes in Computer Science*, Springer, Berlin, Germany, pp. 34–45, 2014.

[28] N. Teslya and S. Savosin, "Matching ontologies with Word2Vec-based neural network," *Computational Science and Its Applications-ICCSA 2019*, Springer, Berlin, Germany, pp. 745–756, 2019.
[29] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, no. 5-6, pp. 907–928, 1995.
[30] G. Bella, F. Giunchiglia, and F. McNeill, "Language and domain aware lightweight ontology matching," *Journal of Web Semantics*, vol. 43, pp. 1–17, 2017.
[31] P. Ochieng and S. Kyanda, "Large-scale ontology matching," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.
[32] L. Otero-Cerdeira, F. J. Rodríguez-Martínez, and A. Gómez-Rodríguez, "Ontology matching: a literature review," *Expert Systems with Applications*, vol. 42, no. 2, pp. 949–971, 2015.
[33] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2011.
[34] J. Barrachina, P. Garrido, M. Fogue et al., "VEACON: a Vehicular Accident Ontology designed to improve safety on the roads," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1891–1900, 2012.
[35] L. Denoyer and P. Gallinari, "The wikipedia XML corpus," *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pp. 12–19, Springer, Berlin, Germany, 2006.
[36] W. Gao, M. R. Farahani, A. Aslam, and S. Hosamani, "Distance learning techniques for ontology similarity measuring and ontology mapping," *Cluster Computing*, vol. 20, no. 2, pp. 959–968, 2017.
[37] C. Fellbaum, "WordNet," *Theory and Applications of Ontology: Computer Applications*, Springer, Berlin, Germany, pp. 231–243, 2010.
[38] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity flooding: a versatile graph matching algorithm and its application to schema matching," in *Proceedings of the 18th International Conference on Data Engineering*, pp. 117–128, San Jose, CA, USA, February 2002.
[39] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," *IIWeb*, vol. 3, pp. 73–78, 2013.
[40] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5526127, 16 pages
https://doi.org/10.1155/2021/5526127

Research Article

Compact Sine Cosine Algorithm with Multigroup and Multistrategy for Dispatching System of Public Transit Vehicles

Minghui Zhu,[1] Shu-Chuan Chu,[1] Qingyong Yang,[1] Wei Li,[2] and Jeng-Shyang Pan[1]

[1]*College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2]*College of Computer Science and Technology, Harbin Engineering University, Harbin150001, China*

Correspondence should be addressed to Jeng-Shyang Pan; jengshyangpan@gmail.com

Received 7 January 2021; Revised 4 March 2021; Accepted 20 March 2021; Published 30 March 2021

Academic Editor: Chi-Hua Chen

This paper studies the problem of intelligence optimization, a fundamental problem in analyzing the optimal solution in a wide spectrum of applications such as transportation and wireless sensor network (WSN). To achieve better optimization capability, we propose a multigroup Multistrategy Compact Sine Cosine Algorithm (MCSCA) by using the compact strategy and grouping strategy, which makes the initialized randomly generated value no longer an individual in the population and avoids falling into the local optimum. New evolution formulas are proposed for the intergroup communication strategy. Performance studies on the CEC2013 benchmark demonstrate the effectiveness of our new approach regarding convergence speed and accuracy. Finally, we apply MCSCA to solve the dispatch system of public transit vehicles. Experimental results show that MCSCA can achieve better optimization.

1. Introduction

The optimization problem is the problem of finding the optimal solution according to the optimization direction in a feasible solution domain defined by constraints [1]. Inspired by existing natural phenomena, a large number of researchers have devoted themselves to heuristic algorithms. Among them, the metaheuristic algorithm which shares the collective information of all individuals based on species behavior is a recent hotspot. Swarm intelligence algorithm is a metaheuristic algorithm [2] that can get better results without consuming too much computing time. Swarm intelligence generally simulates a living population, such as a particle swarm that simulates bird swarm [3, 4], wolf swarm [5], fish swarm [6], cat swarm [7–9], and artificial bee colony [10, 11]. There are also inanimate objects like fireworks [12]. Swarm intelligence optimization algorithms generally have the following characteristics: there are many particles (representing each type of agent), the individuals are independent of each other, individuals move position according to different mechanisms to explore the solution space, and the position movement mechanism generally introduces random numbers for better exploration.

The Sine Cosine Algorithm (SCA) [1] is a new swarm intelligence optimization algorithm proposed by the Australian scholar Mirjalili in 2016. This algorithm has a simple structure and fewer parameters and is easy to implement. In recent years, many scholars have further optimized this algorithm to solve different problems. It is mainly optimized from two aspects. The one aspect is to optimize the algorithm itself, such as the multiobjective version of the SCA algorithm [13] and the binary version of the SCA algorithm [14, 15], based on the opposite learning improved SCA algorithm [16, 17]. The other is to combine the algorithm with other algorithms, such as the SCA algorithm combined with particle swarm optimization algorithm [18, 19] and the SCA algorithm combined with differential evolution algorithm [20–23].

Sine Cosine Algorithm with Multigroup and Multistrategy (MMSCA) is an improved SCA algorithm [24]. MMSCA divides the population into groups and uses different update strategies in different groups. It uses two strategies: rand strategy, and best strategy. In the rand strategy, the next generation of solving goals is chosen by using multiple roulette methods. In the best strategy, the best individual in all populations is still the solution target.

The SCA and the improved SCA algorithms [24–26] are used in many different fields to solve different problems. For example, the SCA algorithm can solve the economic and emission dispatch problem [27, 28], a design damping controller [29], and a system for battery charging [30], predict wind speed [31], reconfigure the power distribution network [32], segment image [33], and control the load frequency of power system [34]. The SCA and improved SCA algorithms still have some disadvantages. A disadvantage of the SCA algorithm is the algorithm that requires evaluation of each solution in the swarm and needs to call the target function multiple times. It increases the complexity of the algorithm. Another disadvantage is the slow convergence speed.

Based on the above discussions, it is key to effectively deal with high complexity and slow convergence speed. Unlike the existing SCA methods, we develop a compact SCA with multigroup and multistrategy, named MCSCA, to make the initialized randomly generated value no longer an individual in the population and avoid falling into the local optimum. Compact strategy [35–38] is a technique based on a probabilistic model. This technique reduces the memory usage and computation time of the algorithm by using a probabilistic model to replace the population from a macroperspective. The idea of grouping [39–41] implies that many populations can be iteratively updated simultaneously. The advantage of this method is that it ensures population diversity and further improves the search capability and performance of the algorithm. In particular, when solving complex optimization problems, grouping populations is an effective way to improve the efficiency and accuracy of the algorithm.

The superiority of the proposed MCSCA is proved by the comparison between MCSCA and several heuristic algorithms on benchmark functions. To test the ability of MCSCA in solving practical problems, we apply it to the dispatching system of public transit vehicles, which is one of the problems that transportation needs to solve, which is related to the economic and social benefits of public transportation companies.

The main contributions of the paper are listed as follows:

(1) We develop a new compact SCA algorithm for intelligent optimization problems

(2) We propose an optimization approach by designing new evolution formulas and utilizing a grouping strategy

(3) We extend the MCSCA algorithm in the application of the dispatching system of public transit vehicles

We conduct extensive empirical studies on the CEC2013 benchmark in Section 4. The empirical studies confirm that our new approach significantly outperforms the state-of-the-art approaches.

The rest of the paper is organized as follows. Section 2 gives a brief retrospect to the SCA algorithm and the basic principles of the compact. Section 3 formally proposes the MCSCA algorithm. In Section 4, the results of numerical experiments are presented and discussed. Section 5 introduces the application of this algorithm in the dispatching system of public transit vehicles. Finally, Section 6 gives a conclusion.

2. Related Works

Besides the related works discussed above, other related works are categorized as follows.

2.1. Sine Cosine Algorithm. The SCA algorithm is a metaheuristic algorithm that uses mathematical equations to estimate the global optimal solution of the optimization problem. It creates multiple initial random candidate solutions and requires them to use mathematical models based on sine and cosine functions to fluctuate outward or toward the best solution. The algorithm also integrates several random and adaptive variables to emphasize the exploration and use of the search space at different stages of optimization. The update formula is as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 \times P_b^t - X_i^t \right|, & r_4 < 0.5, \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 \times P_b^t - X_i^t \right|, & r_4 \geq 0.5, \end{cases} \tag{1}$$

where X_i^t is the position of the i-th dimension in the t-th iteration. r_1, r_2, r_3, and r_4 are random numbers, $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$, and $r_4 \in [0, 1]$, and p_b^t represents the best individual position after t iterations. The updated formula of r_1 is as follows:

$$r_1 = a - a\frac{t}{T}, \tag{2}$$

where a is a constant, t is the number of current iterations, and T is the maximum number of iterations.

2.2. Fundamentals of Compact and Grouping. The essence of the compact is based on a probability model to represent the distribution of all particles. The virtual population replaces the actual solution population. This virtual population is encoded in a data structure. This data structure is named the disturbance vector and represented by PV. There are K particles in the original population, and each particle has n dimensions. After adding the compact, each dimension can be represented by a normal distribution, so that the original $K \times n$ matrix becomes a $2 \times n$ matrix:

$$PV^t = \left[\mu^t, \sigma^t\right], \tag{3}$$

where μ and σ are, respectively, vectors containing, for each design variable, mean and standard deviation values of a Gaussian Probability Distribution Function (PDF) [42] truncated within the interval [-1, 1].

The sampling mechanism of a design variable $x[i]$ associated with a generic candidate solution x from PV requires an extensive explanation. For each design variable indexed by i, a truncated Gaussian PDF with mean value $\mu[i]$ and the standard deviation $\sigma[i]$ is associated.

$$\mathrm{PDF}(\mathrm{truncNorm}(x)) = \frac{e^{-\left(\left((x-\mu[i])^2\right)/\left(2\sigma[i]^2\right)\right)\sqrt{2/\pi}}}{\sigma[i]\left(\mathrm{erf}\left((\mu[i]+1)/\left(\sqrt{2}\,\sigma[i]\right)\right)-\mathrm{erf}\left((\mu[i]-1)/\sqrt{2\sigma[i]}\right)\right)}, \tag{4}$$

where erf is the error function; see [43]. The update rule for μ and σ is

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_p}(\mathrm{winner}[i] - \mathrm{loser}[i]), \tag{5}$$

$$\left(\sigma^{t+1}[i]\right)^2 = \left(\sigma^t[i]\right)^2 + \left(\mu^t[i]\right)^2 - \left(\mu^{t+1}[i]\right)^2 + \frac{1}{N_p}\left(\mathrm{winner}[i]^2 - \mathrm{loser}[i]^2\right), \tag{6}$$

where i represents the i-th dimension, N_p is the virtual population number, and winner and loser are obtained by comparing the two particles generated by PV.

By constructing a Chebyshev polynomial, the probability density function can correspond to a cumulative distribution function with values ranging from 0 to 1. The cumulative distribution function is described by the following equation:

$$\begin{aligned}\mathrm{CDF} &= \int_{-1}^{x} \mathrm{PDF}\,\mathrm{d}x = \int_{-1}^{x} \frac{\sqrt{2/\pi}e^{-\left((x-\mu)^2/2\sigma^2\right)}}{\sigma\mathrm{erf}\left((\mu+1)/\sqrt{2}\,\sigma\right)-\mathrm{erf}\left((\mu-1)/\sqrt{2}\,\sigma\right)}\mathrm{d}x,\\ \mathrm{CDF} &= \frac{\mathrm{erf}\left((\mu+1)/\sqrt{2}\,\sigma\right)+\mathrm{erf}\left((x-\mu)/\sqrt{2}\,\sigma\right)}{\mathrm{erf}\left((\mu+1)/\sqrt{2}\,\sigma\right)-\mathrm{erf}\left((\mu-1)/\sqrt{2}\,\sigma\right)}.\end{aligned} \tag{7}$$

The inverse function of the cumulative distribution is as follows:

$$y = \sqrt{2}\delta\,\mathrm{erf}^{-1}\left(-\mathrm{erf}\left(\frac{\mu+1}{\sqrt{2}\,\delta}\right) - x\mathrm{erf}\left(\frac{\mu-1}{\sqrt{2}\,\delta}\right) + x\mathrm{erf}\left(\frac{\mu+1}{\sqrt{2}\,\delta}\right)\right) + \mu. \tag{8}$$

Many metaheuristic algorithms have applied the idea of grouping, such as parallel PSO [39], parallel ACO [40], and parallel QUATRE [41]. This method changes the basic characteristics of the traditional SCA algorithm. Each group evolves independently, and the entire population adopts a unified mechanism to evolve. Therefore, it can effectively increase the speed of calculation. Different groups adopt different update formulas, which helps prevent falling into the local optimum while preventing premature convergence.

Although the compact strategy reduces memory usage and computation time, it limits the search capability and convergence speed of the algorithm. Therefore, the compact strategy and the group strategy are combined to improve the performance of the algorithm [44].

2.3. Dispatching System of Public Transit Vehicles. There are many issues worth studying in the direction of transportation, such as vehicle speed estimation and prediction [45, 46], traffic monitoring [47], traffic accident prediction [48], bus arrival time prediction [49], and dispatching system of public transit vehicles [50]. Public transit vehicles have the advantages of large passenger capacity, low road traffic conditions, and access to many areas. So, public transit vehicle is an important direction in this field. It has many issues worth studying, such as the choice of bus types, site construction and road selection [51], calculation of greenhouse gas emissions from public transit vehicles [52], integration of unmanned driving, and public transportation.

In this paper, the improved SCA algorithm is applied to the dispatching system of public transit vehicles. Intelligent dispatch of operating vehicles is one of the problems that transportation needs to solve, which is related to the economic and social benefits of public transportation companies. The bus scheduling problem is a complex nonlinear dynamic optimization problem using a bus dispatching model that takes into account the interests of both the company and the passengers. It is to find an optimal solution that satisfies the proposed objective function in the solution space that satisfies the scheduling constraints. Many algorithms have been used for the dispatching system of public transit vehicles, such as ACO [53], GA [54, 55], and Immune Algorithms (IA) [56]. The dispatching system of public transit vehicles still has many problems such as the fact that weights of the multiobjective function are difficult to be determined, the original data source is not accurate, and the algorithm convergence is slow.

3. Compact SCA Algorithm with Multigroup and Multistrategy

In this section, we first propose a compact SCA algorithm in Section 3.1 and then develop optimization techniques inspired by the idea of grouping to avoid the local optimum issue in Section 3.2.

3.1. Compact Sine Cosine Algorithm (CSCA). The pseudocode of our compact SCA algorithm denoted by CSCA is shown in Algorithm 1. We initially set the dimension as d,

the population size as Np, and the upper and lower bounds as ub and lb. The vectors μ and σ of PV are initialized to 0 and 10. The variable best is set as ub, which is used to store the global optimum, and Fmin is set to infinity, which is used to store the fitness value corresponding to the global optimum (Line 1–4). Then, we iteratively use equation (8) to generate a random solution ranging from -1 to 1 and use equation (9) to control the generated solution in the decision space (Line 6). The solution is updated according to equation (1) (Line 7). The variables winner and loser are set as a random solution and updated solution. The values of the objective function corresponding to the two solutions are compared, and winner is set as the better solution (Line 8). After that, for the solution, we first update PV according to equations(5)and(6) (Line 10). The variable $\text{fit}_{\text{winner}}$ represents the objective function value corresponding to winner. If $\text{fit}_{\text{winner}}$ is less than the given threshold, the best and Fmin values will be updated to winner and $\text{fit}_{\text{winner}}$, respectively (Line 11–13). Finally, $\text{fit}_{\text{winner}}$ and Fmin are compared to update the global optimum.

$$x_1 = \frac{x}{2}(\text{ub} - \text{lb}) + \frac{1}{2}(\text{ub} + \text{lb}). \tag{9}$$

3.2. Compact Sine Cosine Algorithm with Multigroup and Multistrategy (MCSCA). The SCA algorithm can generate a set of values per iteration. After adding the idea of compact, the initialized randomly generated value is no longer an individual in the population, but a set of vectors used for compact. In this case, it is easy to fall into the local optimum. Therefore, the idea of grouping is added. Grouping is a common idea in optimization algorithms. After adding group ideas, a set of values can be generated per iteration, which increases the diversity of the population and jumps out of the local optimal. So, the searchability and convergence speed are improved, and the solution is found faster (Algorithm 1).

The communication strategy between groups adopts the idea of differential evolution. The differential evolution algorithm is a random parallel optimization algorithm based on population evolution and is a simple and efficient random global optimization method [57]. This article uses the following differential evolution formula and improved differential evolution formula:

$$\text{DE/best/2: } x = x_{\text{best}} + F(x_{r1} - x_{r2}) + F(x_{r3} - x_{r4}), \tag{10}$$

$$\text{DE/current} - \text{to} - \text{rand/1: } x = x_{\text{current}} + F(x_{\text{current}} - x_{r1}) + F(x_{r2} - x_{r3}), \tag{11}$$

$$\text{DE/best} - \text{to} - \text{rand/1: } x = x_{\text{best}} + F(x_{\text{best}} - x_{r1}) + F(x_{r2} - x_{r3}), \tag{12}$$

$$\text{DE/best} - \text{to} - \text{current/1: } x = x_{\text{best}} + F(x_{\text{best}} - x_{\text{current}}) + F(x_{r1} - x_{r2}), \tag{13}$$

where F stands for mutation factor, generally taking the value 0.5, and x_{r1}, x_{r2}, x_{r3}, and x_{r4} represent random individuals in the population. x_{best} is the global optimal individual and x_{current} represents the current individual.

The original differential evolution strategy is to update the individuals in the population according to the above formula. This algorithm groups the population, and x in the corresponding formula also changes from the individual to the optimal of a group. For example, x_r originally represents a random individual in the population, which represents the optimal individual in a random group in this algorithm. Four formulas are used for communication between groups, among which equations (11)–(13) are improved on common formulas. DE/current-to-rand/1 represents updating the current solution with a random solution, DE/best-to-rand/1 represents updating the global optimum with a random solution, and DE/best-to-current/1 represents updating the global optimum with the current solution. Using the random solution and the current solution to influence the global optimum can speed up the convergence speed, and updating the current solution can increase the randomness of the solution.

The pseudocode of our Compact Sine Cosine Algorithm with Multigroup and Multistrategy denoted by MCSCA is shown in Algorithm 2. We initially set the variable globalbest to represent the global optimum, set the variable globalfmin to represent the fitness value corresponding to the global optimum, set the variable g to represent the number of groups, set $G[i] \cdot \text{best}$ to store the best in the group, and set $G[i] \cdot F\min$ to store the fitness value corresponding to $G[i] \cdot \text{best}$(Line 1–3). Dimension, population size, upper, lower, and vectors of PV are consistent with those set by CSCA. Then, we iteratively use equations (8) and (9) to generate a random solution in the decision space and update this solution according to equation (1) (Line 6–7). winner and loser are set as this random solution and updated solution. The values of the objective function corresponding to the two solutions are compared, and winner is set as the better solution. $\text{fit}_{\text{winner}}$ represents the objective function value corresponding to the winner, and we update PV according to equations (5) and (6) (Line 9).

After that, the optimum in the group and the corresponding fitness value are updated by comparing $\text{fit}_{\text{winner}}$ with $G[i] \cdot F\min$. The global optimum is updated by communication strategy between groups. The specific intergroup communication strategy is to cycle the number of groups, and the following four strategies are executed with equal probability.

Strategy 1 compared the fitness value of the current group and the global optimum to determine whether to

replace it. If there is no need to replace, update the global optimum according to equation (10). Strategy 2 compares the optimal value of the current group and the optimal value of the random group. If the fitness value of the random group is better, the best value and fitness value of the current group are replaced with the best value and fitness value of this random group. Otherwise, the random group is updated according to equation (11). Strategy 3 and Strategy 4 update the global optimum according to equations (12) and (13), respectively. After executing the intergroup strategy, use the sorting statement to find the worst and the best of optimal individuals of all groups and update the worst to the best (Algorithm 2).

4. Experiments

We conduct extensive empirical studies to evaluate the efficiency and effectiveness of our proposed algorithms for the dispatching system of public transit vehicles. In specific, we evaluate the following algorithms:

(1) SCA: the algorithm in [1].

(2) MMSCA: the algorithm in [24].

(3) CSCA: the approach discussed in Section 3.1

(4) MCSCA: the approach discussed in Section 3.2

All algorithms are implemented in MATLAB2018a. All experiments are conducted on a machine with an Intel (R) Core (TM) 2.60 GHz CPU and 12 GB memory running 64-bit Windows 10. We evaluate the performance of all algorithms on the CEC2013 benchmark functions as follows:

(1) $\mathbf{f}_1 - \mathbf{f}_5$ are unimodal benchmark functions

(2) $\mathbf{f}_6 - \mathbf{f}_{20}$ are basic multimodal functions

(3) $\mathbf{f}_{21} - \mathbf{f}_{28}$ are composite functions

There are four key parameters in the intergroup communication strategy of the MCSCA algorithm, that is, the mutation factor corresponding to the four formulas. The value of the mutation factor will affect the convergence of the algorithm. If the value of the mutation factor is too small, it will easily fall into the local optimum, and if the value of the mutation factor is too large, the algorithm is difficult to converge. So, Taguchi's method [58] is used to obtain a reasonable combination of the four mutation factors. Taguchi's method detects part of the possible combination of factors, but not all combinations. This method uses the smallest number of trials to detect the best combination. The value of F ranges from 0 to 2. So, the level settings of the four factors are 0.5, 1, and 1.5. A full-factorial analysis needs $3^4 = 81$ experiments. Taguchi's method adopts the orthogonal arrays. Therefore, an orthogonal array L_9, (3^4) that contains only 9 experiments is adopted in our experiment. We conducted ten tests on CEC2013 to find the average value. When the mutation factors are all 0.5, the effect is the best.

4.1. Comparing CSCA and MCSCA with SCA and MMSCA. We compare our proposed algorithms, CSCA and MCSCA, with the existing algorithms, SCA and MMSCA. Here, we selected 15 figures with obvious optimization effects to show in Figure 1. Table 1 presents the results in the 28 CEC2013 benchmark functions. Generally, MCSCA outperforms CSCA; the only difference between these two algorithms is that MCSCA uses the proposed Multigroup and Multistrategy. Thanks to the compact technique, the performance of CSCA is much better than the performance of SCA in most functions.

As shown in Table 1, compared with the SCA algorithm, the MCSCA achieves better results in 16 benchmark functions. Compared with MMSCA, the proposed MCSCA algorithm performs better on 17 benchmark test functions. Compared with CSCA, the MCSCA algorithm is better in all benchmark functions. Among the unimodal benchmark functions, our MCSCA approach over $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_5$ can achieve the best accuracy compared to that of SCA and MMSCA algorithms. From Figure 1, we can see that the convergence speed in our MCSCA algorithm increases significantly. Among the eight composite functions, for $\mathbf{f}_{26}$ and $\mathbf{f}_{28}$, the MMSCA algorithm is better optimized. For the remaining six composite functions, the MCSCA algorithm is better optimized. The reason for the significant optimization effect is the intergroup communication strategy of MCSCA, which uses the idea of differential evolution. The basic idea of the communication strategy between groups is to use the difference between the two group optimal values to affect the current group optimal value or the global optimal value. It increases the randomness of the solution. After intergroup communication is complete, the worst individual in all groups is found by a ranking statement and replaced with the global optimum. This method speeds up convergence.

4.2. Comparison with the Existing Compact Methods. To further verify the effectiveness of the algorithm, this paper also compares it with three other compact algorithms, including CBA, PCABC, and CPSO. The number of iterations is one thousand. Each algorithm is tested 30 times for the average. The experimental results are shown in Table 2.

It can be seen that the MCSCA algorithm has better solution accuracy compared to CBA, CPSO, and PCABC, except for $\mathbf{f}_1$. CBA and CPSO approach only add compact ideas to the original SCA algorithm, which saves memory but is easy to fall into a local optimum and converge ahead of time. The PCABC, like the MCSCA algorithm, adds grouping ideas to the compact algorithm.

But the intergroup communication strategy of the MCSCA algorithm uses the idea of differential evolution and multiple strategies. It updates the group optimum to increase the diversity of population solutions and updates the global optimum to maintain the strong local optimization ability of the original algorithm and avoid the instability of the algorithm caused by the introduction of the rand strategy. Thus, the MCSCA algorithm has a more pronounced optimization effect.

According to the comparison results of the two phases of experiments, the MCSCA algorithm shows strong competitiveness compared with other algorithms of the same type. To further study the feasibility and effectiveness of this

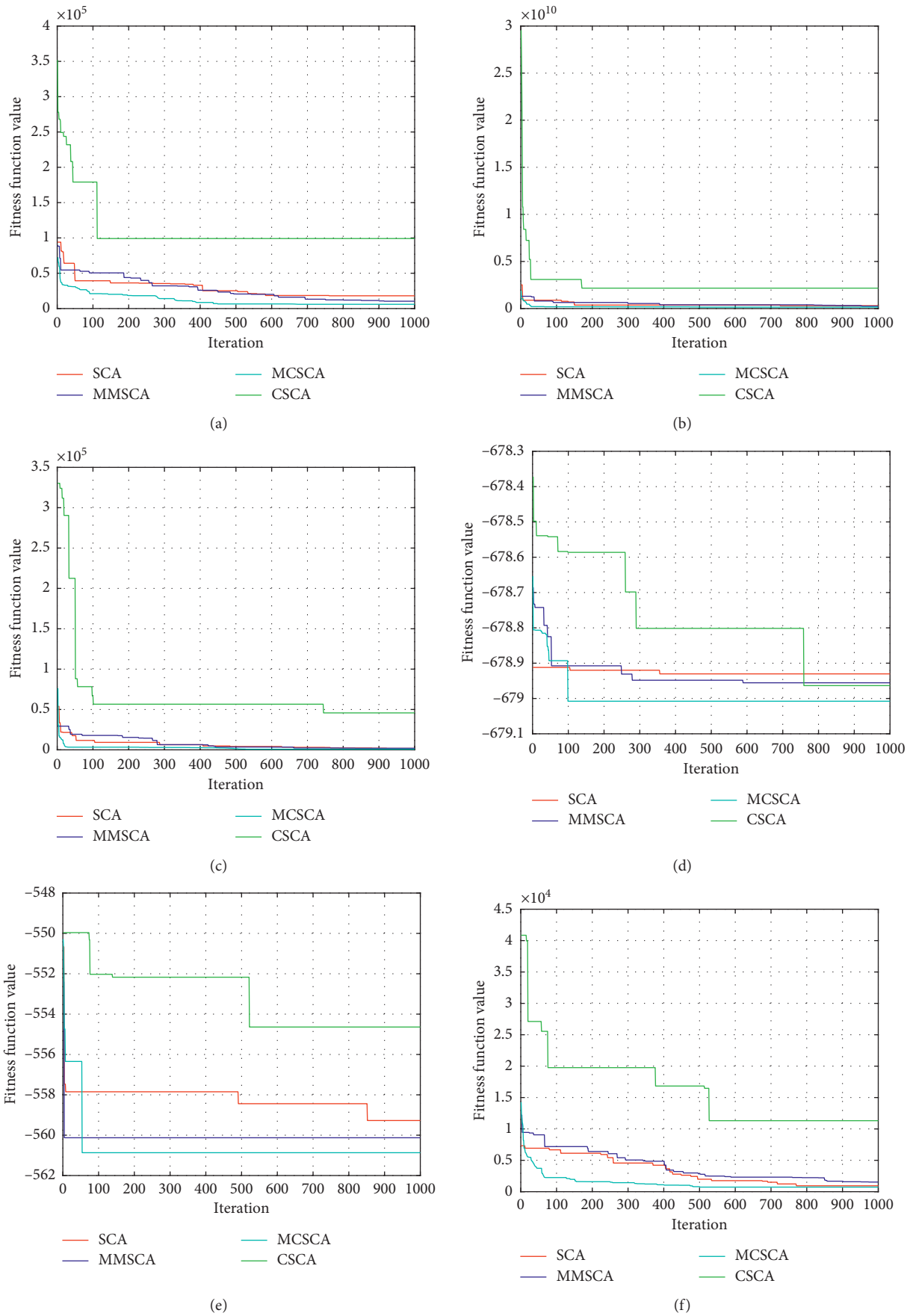


Figure 1: Continued.

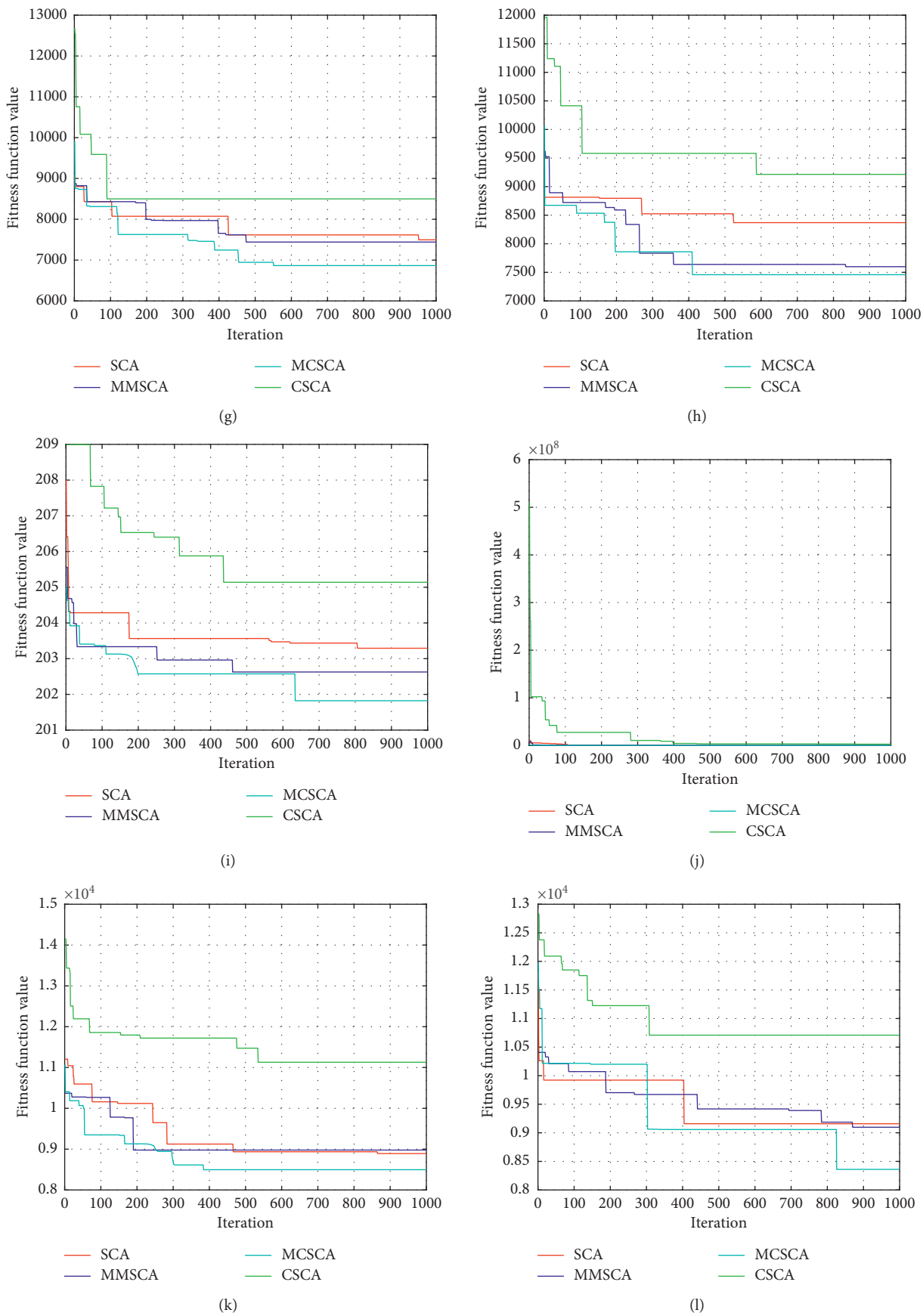


Figure 1: Continued.

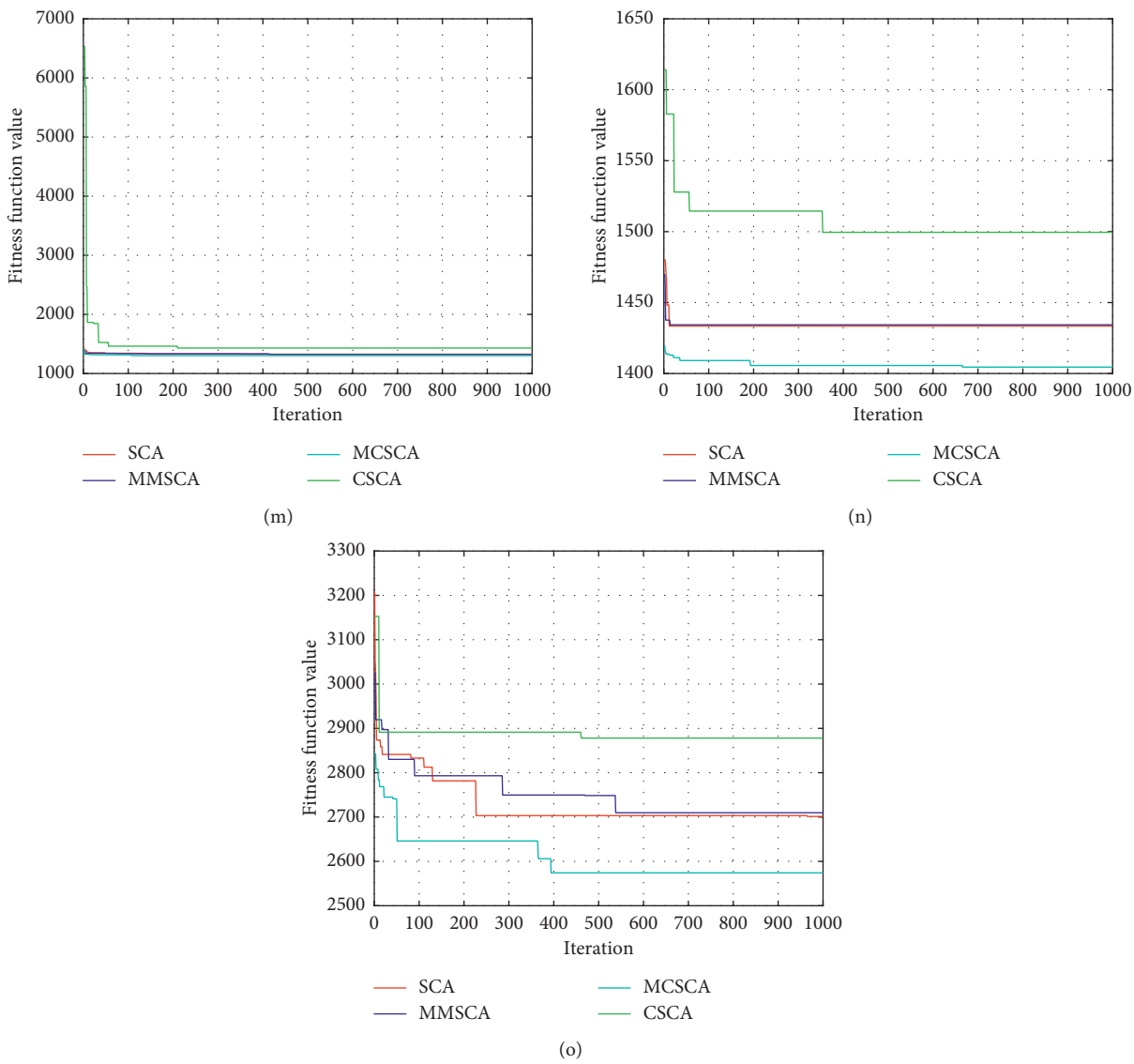


Figure 1: Convergence plots for optimization in the CEC2013 benchmark functions. (a) $\mathbf{f}_1$, (b) $\mathbf{f}_2$, (c) $\mathbf{f}_5$, (d) $\mathbf{f}_8$, (e) $\mathbf{f}_9$, (f) $\mathbf{f}_{10}$, (g) $\mathbf{f}_{14}$, (h) $\mathbf{f}_{15}$, (i) $\mathbf{f}_{16}$, (j) $\mathbf{f}_{19}$, (k) $\mathbf{f}_{22}$, (l) $\mathbf{f}_{23}$, (m) $\mathbf{f}_{24}$, (n) $\mathbf{f}_{25}$, and (o) $\mathbf{f}_{27}$.

Input: **Parameters** d, N_p, ub, **and** lb
Output: **Global optimum** best **and its fitness value** F min
for $i = 1{:}d$ **do**
  initialize $\mu = 0$, $\sigma = \lambda = 10$;
  initialize best = ub, F min = inf ;
end for
while t<Max Generation **do**
  Get x_1 from PV via equations (8), (9);
  Update x_1 to get x_2 via equation (1);
  [winner, loser] = compete (x_1 ,x_2);
  for i = 1:d **do**
    Update PV via equations (5), (6);
    if $\text{fit}_{\text{winner}} < \text{Fmin}$ **then**
    best = winner; F min = $\text{fit}_{\text{winner}}$;
    end if
  end for
end while

Algorithm 1: CSCA.

```
Input: Parameters d, N_p, ub , lb, and g
Output: Global optimum globalbest and its fitness value globalfmin
Set the number of groups g, each group is G_i;
initialize G[i] · PV , G[i] · best and G[i] · F min of each group;
Initialize globalbest = G[1] · best, globalfmin = G[1] · F min;
while t<Max Gerneration do
  for i = 1:g do
    Get x_1 , x_2 via equations (8). (9) and (1);
    [winner, loser, fit_winner] = compete (x_1 ,x_2 );
    for i = 1:d do
      Update G[i] · PV via equations (5), (6);
    end for
    if fit_winner <G[i] · F min then
      G[i] · best = winner; G[i] · F min = fit_winner;
    end if
    Update globalbest and globalfmin;
    if rand<0.25 then
      Execute Strategy 1;
    else if rand<0.5 then
      Execute Strategy 2;
    else if rand<0.75 then
      Execute Strategy 3;
    else
      Execute Strategy 4;
    end if
  end for
  Change the worst individual to the best individual by the sorting statement;
end while
```

Algorithm 2: MCSCA.

algorithm in practical applications, this paper applies the algorithm to the dispatching system of public transit vehicles.

4.3. Comparing MCSCA with DE and DE Variants. The intergroup communication strategy of the proposed algorithm uses a differential evolution mechanism. So, in order to fully demonstrate the effectiveness of the proposed method, this paper also compares the proposed algorithm with DE and DE variants, including JADE, SHADE, and LSHADE [59–61]. The test results are shown in Table 3. JADE algorithm incorporates parameter adaptation and greedy compilation strategies in the traditional DE algorithm. SHADE uses a historical memory of recently successful parameter sets based on JADE. LSHADE linearly reduces the population size based on SHADE.

According to the data in Table 3, compared with the DE algorithm and improved DE algorithms, the MCSCA algorithm has better results than these algorithms in 13 functions. But the effect on other benchmark functions is not the best. Compared with the original DE algorithm, the MCSCA algorithm is only unsatisfactory in the effects of f_{17}. The main reason for this effect is that the MCSCA algorithm divides the population into groups and adopts a variety of intragroup communication strategies. Different communication strategies focus on different goals. This method can accelerate the convergence speed without falling into the local optimum while reducing memory.

5. MCSCA Algorithm on the Dispatching System of Public Transit Vehicles

Public transportation is an important part of urban transportation. Good bus dispatching is of great significance for improving the urban transportation environment, improving the travel conditions of citizens, and improving the economic and social benefits of bus companies. Optimal bus dispatching is a rhythmic and repetitive driving plan. It organizes a large number of vehicles on a prescribed route and is made according to the number, direction, and time of passenger flow after studying the rules of passenger flow. It is a complex nonlinear dynamic optimization problem.

Before describing the optimal scheduling of buses, the following assumptions need to be made:

(1) The application is aimed at the dispatch of public transportation vehicles on a certain route in the urban public transportation system

(2) The model is built considering only the one-way case of this route

(3) The buses of this route are of the same vehicle type

(4) When each bus of this route passes through various stations, there are no passengers left

Table 1: Performance evaluation for CSCA, MCSCA, SCA, and MMSCA over CEC2013.

Function	MCSCA	MMSCA	SCA	CSCA
f_1	**1.0012e + 04**	1.1018e + 04	1.5563e + 04	1.1834e + 05
f_2	**1.5879e + 08**	1.7454e + 08	2.4311e + 08	2.1502e + 09
f_3	8.9088e + 12	5.9098e + 10	**5.3981e + 10**	1.2261e + 17
f_4	4.6156e + 04	4.6370e + 04	**4.5172e + 04**	2.7359e + 05
f_5	**1.8310e + 03**	2.0867e + 03	2.1676e + 03	4.1026e + 04
f_6	278.9563	**15.6414**	206.0107	2.5995e + 04
f_7	320.5652	**−605.5122**	−533.9551	1.3614e + 06
f_8	**−679.0443**	−679.0068	−678.9740	−678.8082
f_9	**−560.8198**	−560.1837	−558.7135	−554.2032
f_{10}	**1.2036e + 03**	1.2840e + 03	1.5836e + 03	1.5415e + 04
f_{11}	90.0472	**−10.3529**	12.7879	928.8658
f_{12}	227.4888	**85.8620**	106.4320	1.3068e + 03
f_{13}	321.5240	**193.1019**	196.4991	1.4176e + 03
f_{14}	**5.9463e + 03**	7.2369e + 03	7.2578e + 03	9.1906e + 03
f_{15}	**6.8223e + 03**	7.5356e + 03	7.9090e + 03	9.3590e + 03
f_{16}	**201.7178**	202.8841	202.9772	204.9818
f_{17}	944.1183	**819.8032**	876.3162	3.7177e + 03
f_{18}	1.0708e + 03	**935.6093**	949.7953	4.2394e + 03
f_{19}	**3.6962e + 03**	7.9335e + 03	2.0738e + 04	3.1437e + 06
f_{20}	614.7672	614.4340	**614.3648**	615
f_{21}	**2.1325e + 03**	2.6757e + 03	2.7754e + 03	9.3564e + 03
f_{22}	**7.6664e + 03**	8.5483e + 03	8.7796e + 03	1.1109e + 04
f_{23}	**8.3404e + 03**	9.0079e + 03	8.9873e + 03	1.0256e + 04
f_{24}	**1.3035e + 03**	1.3196e + 03	1.3216e + 03	1.3885e + 03
f_{25}	**1.3992e + 03**	1.4321e + 03	1.4338e + 03	1.4979e + 03
f_{26}	1.5405e + 03	**1.4144e + 03**	1.4243e + 03	1.6308e + 03
f_{27}	**2.6168e + 03**	2.7104e + 03	2.7086e + 03	2.8722e + 03
f_{28}	5.2591e + 03	**4.2623e + 03**	4.4768e + 03	1.3878e + 04

(5) In the same time period, two adjacent buses have the same time interval

(6) Passengers arriving at the station at each time period obey even distribution

(7) The bus has a certain running time between two adjacent stops

(8) The ticket price is the same throughout the journey

Under the above assumptions, the optimal dispatching problem of a certain route is described as follows.

In a dispatching problem of public transportation, it is known that a bus route with a total mileage of L has J stations in total, and the operating time of vehicles in one day is $[t_{\text{early}}, t_{\text{night}}]$. The operating time can be divided into K time periods. The departure interval of the k-th time period is Δt_k. The bus models of this route are the same and arrive at the station on time. The bus fare for each passenger is n. The problem needs to consider both the operating profit of the bus company and the service level of the bus company (the length of waiting time for passengers). According to the passenger flow and operating conditions of each station in a day, it solves the vehicle operating timetable of this route, that is, the departure time interval of each period of operating time.

The purpose of the optimal bus dispatching system is to balance the interests of both the bus company and the passengers, from the perspective of the bus company and the passengers. From the perspective of the bus company, in order to ensure the lowest operating cost of the bus company, it is necessary to control the number of departures of the bus company. From the perspective of the passengers, it is necessary to consider that the long waiting time of the passengers will make the passengers impatient. Therefore, the following objective function is established:

$$F(\Delta t_k) = \alpha C_1 L \sum_{k=1}^{k} \frac{T_k}{\Delta t_k} + \beta C_2 \sum_{k=1}^{k} \sum_{j=1}^{J} m_k \left(\frac{\rho_{kj} \Delta t_k^2}{2} \right). \quad (14)$$

The objective function can be written as

$$F(\Delta t_k) = \alpha f_1(\Delta t_k) + \beta f_2(\Delta t_k), \quad (15)$$

$$f_1(\Delta t_k) = C_1 L \sum_{k=1}^{k} \frac{T_k}{\Delta t_k}, \quad (16)$$

$$f_2(\Delta t_k) = C_2 \sum_{k=1}^{k} \sum_{j=1}^{J} m_k \left(\frac{\rho_{kj} \Delta t_k^2}{2} \right), \quad (17)$$

where α is the weighting coefficient of the consumption of the bus company, β is the weighting coefficient of the consumption of the passengers, $f_1(\Delta t_k)$ represents the objective function cost of the bus company, and $f_2(\Delta t_k)$ represents the objective function of passenger loss. In equation (16), L is the total kilometers of the route, K is the number of time period, k is the time period, T_k represents the time length of the period, Δt_k is the departure interval of the k period, and C_1 is the cost of consumption per bus per kilometer. In equation (17), J is the number of stations, j is the station, and C_2 is the cost of loss per passenger waiting for a unit of time. m_k is the total number of departures in the k-th period and it is equal to the ratio of the length of the period to the departure interval. ρ_{kj} is the passenger arrival rate at the j-th station in the k-th period, and it is equal to the ratio of the number of passengers on the j-th station in the k-th period to the length of time.

Table 2: Performance evaluation for MCSCA, CBA, PCABC, and CPSO over CEC2013.

Function	MCSCA	CBA	PCABC	CPSO
$\mathbf{f}_1$	1.0012e + 04	5.8592e + 04	5.4265e + 04	**8.2174e + 03**
$\mathbf{f}_2$	**1.5879e + 08**	4.6005e + 08	7.1326e + 08	3.4631e + 08
$\mathbf{f}_3$	**8.9088e + 12**	3.4373e + 21	2.7048e + 15	1.9944e + 15
$\mathbf{f}_4$	**4.6156e + 04**	6.8506e + 04	1.0192e + 05	1.0063e + 05
$\mathbf{f}_5$	**1.8310e + 03**	5.4879e + 03	1.5106e + 04	1.5106e + 04
$\mathbf{f}_6$	**278.9563**	1.8592e + 04	7.4003e + 03	5.5797e + 03
$\mathbf{f}_7$	**320.5652**	2.8640e + 04	3.5116e + 04	576.36333
$\mathbf{f}_8$	**−679.0443**	−678.7539	−678.9390	−678.7612
$\mathbf{f}_9$	**−560.8198**	−542.1405	−557.5668	−556.9122
$\mathbf{f}_{10}$	**1.2036e + 03**	1.1691e + 04	6.9226e + 03	1.7629e + 03
$\mathbf{f}_{11}$	**90.0472**	598.9505	468.3890	325.8092
$\mathbf{f}_{12}$	**227.4888**	675.6116	600.3618	402.0084
$\mathbf{f}_{13}$	**321.5240**	938.8861	702.9930	496.6388
$\mathbf{f}_{14}$	**5.9463e + 03**	6.6365e + 03	8.0827e + 03	8.3116e + 03
$\mathbf{f}_{15}$	**6.8223e + 03**	6.1357e + 03	8.2723e + 03	9.0474e + 03
$\mathbf{f}_{16}$	**201.7178**	203.1301	203.3737	204.0167
$\mathbf{f}_{17}$	**944.1183**	1.2152e + 03	1.3819e + 03	1.1049e + 03
$\mathbf{f}_{18}$	**1.0708e + 03**	1.2951e + 03	1.4863e + 03	1.2152e + 03
$\mathbf{f}_{19}$	**3.6962e + 03**	7.5477e + 05	1.1702e + 06	1.2111e + 06
$\mathbf{f}_{20}$	**614.7672**	615	614.9994	615
$\mathbf{f}_{21}$	**2.1325e + 03**	3.2044e + 03	3.4450e + 03	2.7068e + 03
$\mathbf{f}_{22}$	**7.6664e + 03**	1.1060e + 04	9.5938e + 03	1.0378e + 03
$\mathbf{f}_{23}$	**8.3404e + 03**	9.6684e + 03	9.6065e + 03	1.0629e + 03
$\mathbf{f}_{24}$	**1.3035e + 03**	1.8042e + 03	1.3712e + 03	1.3198e + 03
$\mathbf{f}_{25}$	**1.3992e + 03**	1.5600e + 03	1.4826e + 03	1.4109e + 03
$\mathbf{f}_{26}$	**1.5405e + 03**	2.3145e + 03	1.4961e + 03	1.5641e + 03
$\mathbf{f}_{27}$	**2.6168e + 03**	4.3918e + 03	2.8685e + 03	2.7369e + 03
$\mathbf{f}_{28}$	**5.2591e + 03**	1.0705e + 04	7.6122e + 04	7.9081e + 04

Besides, there is a hidden constraint. If the bus company wants to achieve a profitable goal, it must make the total fare collected by the bus company greater than the bus company's minimum consumption cost. That is, the following formula must be satisfied:

$$n \times \frac{\sum_{k=1}^{K} \sum_{j=1}^{J} u_{kj}}{\sum_{k=1}^{K} (T_k/\Delta t_k) > C_1 L,} \tag{18}$$

where u_{kj} is the passenger arrival rate at the j-th station in the k-th period.

The code part of the constraint condition adopts the penalty function method. The penalty function is a commonly used method to deal with constraint conditions. It transforms constrained optimization problems into unconstrained optimization problems. If the fare collected by the bus company is less than its minimum consumption cost, the departure interval is increased, so that the consumption of the bus company is reduced. The departure interval is in minutes. If the condition is not met in one cycle, the time interval of a time period is randomly selected again and incremented by 1.

Assuming that there are 10 stations on the bus route, the total kilometers of the route is 8 km and the operating hours of the bus company are from 6 am to 9 pm every day. The variable x represents the departure interval. The passenger flow of each station in each time period is shown in Table 4.

The other parameters of the MCSCA algorithm are set as follows: $C_1 = 1$, $C_2 = 1$, $n = 1$, and the max of iteration equals 200. This application performs 10 optimization tests for the settings of 3 different parameters α and β and compares with SCA and MMSCA algorithms.

Situation 1. Let $\alpha = 0.2$, $\beta = 0.8$, the operation result is $\min(F(\Delta t_k)) = 3037.2$, $\Delta t_k = (2, 3, 4, 2, 3)$, and the unit is the minute. We obtained the optimized results' comparison as shown in Table 5 and curves of the objective function value as shown in Figure 2.

Situation 2. Let $\alpha = 0.5$, $\beta = 0.5$, the operation result is $\min(F(\Delta t_k)) = 3506.75$, $\Delta t_k = (2, 2, 3, 2, 4)$, and the unit is the minute. We obtained the optimized results' comparison as shown in Table 6 and curves of the objective function value as shown in Figure 3.

Situation 3. Let $\alpha = 0.8$, $\beta = 0.2$, the operation result is $\min(F(\Delta t_k)) = 2743.4033$, $\Delta t_k = (3, 4, 6, 3, 7)$, and the unit is the minute. We obtained the optimized results' comparison as shown in Table 7 and curves of the objective function value as shown in Figure 4.

The simulation result shows that, in the morning peak and evening peak, the passenger flow is large, and the departure frequency is higher, and when the passenger flow is low, the departure frequency is also reduced. The optimization result basically conforms to the actual situation. In actual application, the value of α and β depends on whether the decision-maker is more inclined to benefit the bus company or the consumer. The purpose

Table 3: Performance evaluation for MCSCA, DE, JADE, SHADE, and LSHADE over CEC2013.

Function	MCSCA	DE	JADE	SHADE	LSHADE
f_1	**1.0012e + 04**	1.8334e + 04	1.6936e + 04	1.3601e + 04	1.1211e + 04
f_2	1.5879e + 08	2.2830e + 08	2.4714e + 08	1.8775e + 08	**1.5016e + 08**
f_3	8.9088e + 12	2.0809e + 12	2.0411e + 11	**1.5216e + 11**	9.3578e + 11
f_4	**4.6156e + 04**	5.3020e + 04	5.5735e + 04	4.6472e + 04	5.1199e + 04
f_5	1.8310e + 03	3.8654e + 03	3.5120e + 03	2.7086e + 03	**1.7677e + 03**
f_6	**278.9563**	1.4409e + 03	1.0112e + 03	1.2955e + 03	351.1929
f_7	320.5652	2.2144e + 03	581.7276	**204.7736**	235.5335
f_8	**−679.0443**	−678.9891	−679.0055	−679.0331	−679.0418
f_9	−560.8198	−560.0210	−559.3174	**−561.1851**	−561.0092
f_{10}	**1.2036e + 03**	2.1623e + 03	2.1844e + 03	1.5209e + 03	1.8154e + 03
f_{11}	90.0472	120.7334	114.3473	**73.1779**	80.3474
f_{12}	227.4888	246.6691	247.9068	182.5079	**173.5435**
f_{13}	321.5240	334.2122	292.0396	**283.9232**	330.5799
f_{14}	**5.9463e + 03**	7.1437e + 03	7.2404e + 03	6.5639e + 03	6.5771e + 03
f_{15}	**6.8223e + 03**	7.3093e + 03	7.5481e + 03	7.4742e + 03	7.6433e + 03
f_{16}	**201.7178**	202.6536	202.5590	202.4048	202.3882
f_{17}	944.1183	880.5067	**828.7338**	893.9824	908.6225
f_{18}	1.0708e + 03	1.0352e + 03	1.0004e + 03	1.0181e + 03	**986.8915**
f_{19}	**3.6962e + 03**	7.3193e + 03	9.0551e + 03	6.2207e + 03	6.2646e + 03
f_{20}	614.7672	614.8392	614.74552	**614.4299**	614.6522
f_{21}	**2.1325e + 03**	2.6780e + 03	2.5064e + 03	2.3300e + 03	2.2905e + 03
f_{22}	**7.6664e + 03**	8.5134e + 03	8.8210e + 03	8.1707e + 03	8.5568e + 03
f_{23}	**8.3404e + 03**	9.1687e + 03	9.0751e + 03	8.9719e + 03	8.9474e + 03
f_{24}	1.3035e + 03	1.3004e + 03	1.3059e + 03	1.3007e + 03	**1.2964e + 03**
f_{25}	1.3992e + 03	1.4012e + 03	1.4021e + 03	1.3996e + 03	**1.3985e + 03**
f_{26}	1.5405e + 03	1.5791e + 03	1.5341e + 03	1.5403e + 03	**1.5075e + 03**
f_{27}	2.6168e + 03	2.6780e + 03	2.6312e + 03	2.6169e + 03	**2.6157e + 03**
f_{28}	**5.2591e + 03**	8.5134e + 03	5.3486e + 03	5.5304e + 03	5.3490e + 03

Table 4: Number of passengers at various time sections and various stations.

Time	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
6 : 00–8:30	506	168	417	209	26	23	20	19	10	2
8 : 30–12 : 00	330	165	187	174	67	66	141	140	40	12
12 : 00–16 : 00	127	64	60	58	116	110	158	132	20	6
16 : 00–19 : 00	344	172	254	224	177	178	162	150	70	24
19 : 00–21 : 00	60	32	45	43	17	14	45	42	15	3

Table 5: Optimized results' comparison in $\alpha = 0.2$ and $\beta = 0.8$.

Algorithm	Average value	Best value	Worst value
MCSCA	3155.6667	2902.8	3408.4
SCA	3624.5055	2902.8	4198.8
MMSCA	3216.8	2862.8	3575.2

of designing three sets of different weight values is to prove that MCSCA has a better optimization effect in any case.

According to Tables 5–7, regardless of the values of α and β, the average value of the convergence accuracy of the MCSCA algorithm on this application is significantly better than the SCA algorithm and the MMSCA algorithm. And the difference between the best value and the worst value is significantly smaller than the other two algorithms. It indicates that the MCSCA is more stable. According to Figures 2–4, not only has the MCSCA algorithm a significant advantage in convergence accuracy but also the convergence speed is significantly better than the SCA algorithm and the MMSCA algorithm.

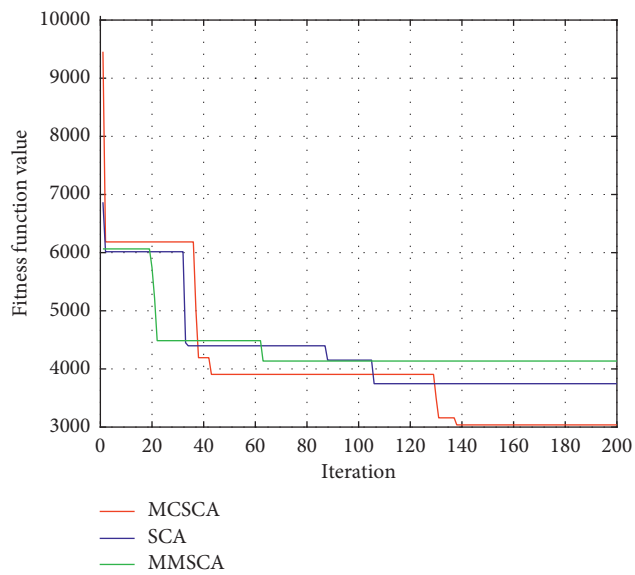


Figure 2: Curves of objective function F value in $\alpha = 0.2$, $\beta = 0.8$.

Table 6: Optimized results' comparison in $\alpha = 0.2$ and $\beta = 0.8$.

Algorithm	Average value	Best value	Worst value
MCSCA	3562.9946	3474	3671.4464
SCA	3678.7678	3474	3987
MMSCA	3601.9887	3525	3662.5

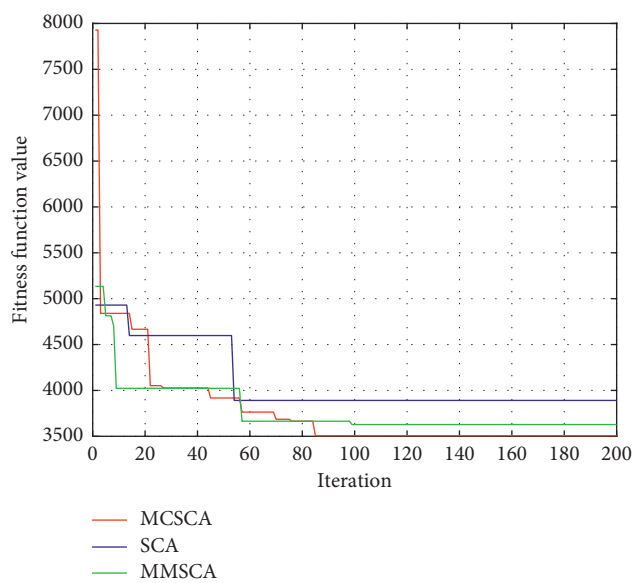


Figure 3: Curves of objective function F value in $\alpha = 0.5$, $\beta = 0.5$.

Table 7: Optimized results' comparison in $\alpha = 0.2$ and $\beta = 0.8$.

Algorithm	Average value	Best value	Worst value
MCSCA	3562.9946	3474	3671.4464
SCA	3678.7678	3474	3987
MMSCA	3601.9887	3525	3662.5

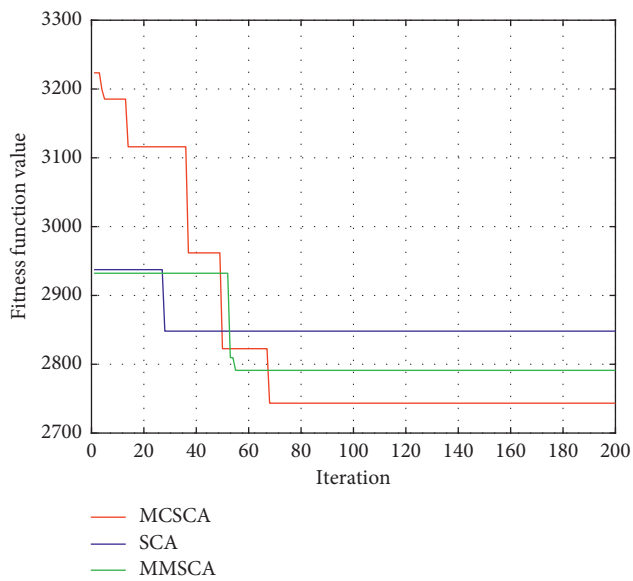


Figure 4: Curves of objective function F value in $\alpha = 0.8$, $\beta = 0.2$.

6. Conclusion

In this paper, we developed a compact-based SCA algorithm, CSCA, which can reduce the number of optimized variables and the required running memory. To improve the performance of our approach, we further proposed the MCSCA algorithm, which combines grouping strategy to speed up the convergence speed. Extensive empirical studies on benchmark CEC2013 demonstrate the effectiveness of our approach and the efficiency of our techniques. Furthermore, applying the MCSCA algorithm to the dispatching system of public transit vehicles can get a minimum cost value than that of SCA and MMSCA algorithms, which further testify the effectiveness and superiority of the proposed MCSCA algorithm. A possible future work is considered to continue the improvement of the SCA algorithm and apply it to more practical engineering problems.

Data Availability

Public CEC2013 was used. No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 809–818, 2016.

[2] S. C. Chu, H. C. Huang, J. F. Roddick et al., "Overview of algorithms for swarm intelligence," in *Proceedings of the International Conference on Computational Collective Intelligence*, pp. 28–41, Da Nang, Vietnam, November 2011.

[3] C.-l. Sun, J.-c. Zeng, and J.-s. Pan, "An improved vector particle swarm optimization for constrained optimization problems," *Information Sciences*, vol. 181, no. 6, pp. 1153–1163, 2011.

[4] W. Hao, M. Liang, C. Sun et al., "Multiple-strategy learning particle swarm optimization for large-scale optimization problems," *Complex & Intelligent Systems*, vol. 7, 2020.

[5] P. Hu, J. S. Pan, and S. C. Chu, "Improved binary grey wolf optimizer and Its application for feature selection," *Knowledge-Based Systems*, p. 105746, 2020.

[6] Q. W. Chai, S. C. Chu, J. S. Pan et al., "Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-D terrain," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 2, pp. 90–102, 2020.

[7] S. C. Chu, P. W. Tsai, and J. S. Pan, "Cat swarm optimization," in *Proceedings of the Pacific Rim International Conference on Artificial intelligence*, pp. 854–858, Berlin, Germany, August 2006.

[8] P. W. Tsai, J. S. Pan, S. M. Chen et al., "Parallel cat swarm optimization," in *Proceedings of the 2008 international conference on machine learning and cybernetics*, pp. 3328–3333, Kunming, China, July 2008.

[9] P.-W. Tsai, J.-S. Pan, S.-M. Chen, and B.-Y. Liao, "Enhanced parallel cat swarm optimization based on the Taguchi method," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.

[10] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-s. Pan, "Multi-strategy ensemble artificial bee colony algorithm," *Information Sciences*, vol. 279, pp. 587–603, 2014.

[11] P. W. Tsai, J. S. Pan, B. Y. Liao et al., "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5081–5092, 2009.

[12] Y. Tan, C. Yu, S. Zheng, and K. Ding, "Introduction to fireworks algorithm," *International Journal of Swarm Intelligence Research*, vol. 4, no. 4, pp. 39–70, 2013.

[13] M. A. Tawhid and V. Savsani, "Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems," *Neural Computing and Applications*, vol. 31, no. 2, pp. 915–929, 2019.

[14] A. I. Hafez, H. M. Zawbaa, E. Emary et al., "Sine cosine optimization algorithm for feature selection," in *Proceedings of the 2016 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–5, Sinaia, Romania, August 2016.

[15] K. S. Reddy, L. K. Panwar, B. Panigrahi, and R. Kumar, "A new binary variant of sine-cosine algorithm: development and application to solve profit-based unit commitment problem," *Arabian Journal for Science and Engineering*, vol. 43, no. 8, pp. 4041–4056, 2018.

[16] M. Abd Elaziz, D. Oliva, and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Systems with Applications*, vol. 90, pp. 484–500, 2017.

[17] D. Bairathi and D. Gopalani, "Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks," in *Proceedings of the 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 438–444, Jaipur, India, December 2017.

[18] K. Chen, F. Zhou, L. Yin, S. Wang, Y. Wang, and F. Wan, "A hybrid particle swarm optimizer with sine cosine acceleration coefficients," *Information Sciences*, vol. 422, pp. 218–241, 2018.

[19] M. Issa, A. E. Hassanien, D. Oliva, A. Helmi, I. Ziedan, and A. Alzohairy, "ASCA-PSO: adaptive sine cosine optimization

algorithm integrated with particle swarm for pairwise local sequence alignment," *Expert Systems with Applications*, vol. 99, pp. 56–70, 2018.

[20] S. Bureerat and N. Pholdee, "Adaptive sine cosine algorithm integrated with differential evolution for structural damage detection," in *Proceedings of the International Conference on Computational Science and its Applications*, pp. 71–86, Trieste, Italy, July 2017.

[21] M. Abd Elaziz, A. A. Ewees, D. Oliva et al., "A hybrid method of sine cosine algorithm and differential evolution for feature selection," in *Proceedings of the International Conference on Neural information Processing*, pp. 145–155, Guangzhou, China, July 2017.

[22] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing*, vol. 62, pp. 1019–1043, 2018.

[23] C. Zhou, L. Chen, Z. Chen, X. Li, and G. Dai, "A sine cosine mutation based differential evolution algorithm for solving node location problem," *International Journal of Wireless and Mobile Computing*, vol. 13, no. 3, pp. 253–259, 2017.

[24] Q. Yang, S. C. Chu, J. S. Pan et al., "Sine cosine algorithm with multigroup and multistrategy for solving CVRP," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8184254, 10 pages, 2020.

[25] M. Abd Elfattah, S. Abuelenin, A. E. Hassanien et al., "Handwritten Arabic manuscript image binarization using sine cosine optimization algorithm," in *Proceedings of the International Conference on Genetic and Evolutionary Computing*, pp. 273–280, Fuzhou, China, November 2016.

[26] J. S. Pan, T. T. Nguyen, A. H. Vu et al., "A new optimization based on flower and sine cosine for saving fuel consumption problem," *Journal of Computers and Applied Science Education*, vol. 5, no. 3, 2018.

[27] E. Mostafa, M. Abdel-Nasser, and K. Mahmoud, "Performance evaluation of metaheuristic optimization methods with mutation operators for combined economic and emission dispatch," in *Proceedings of the 2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, pp. 1004–1009, Cairo, Egypt, December 2017.

[28] P. P. Singh, R. Bains, G. Singh et al., "Comparative analysis on economic load dispatch problem optimization using moth flame optimization and sine cosine algorithms," *International Journal Of Advance Research And Innovative Ideas In Education*, vol. 3, no. 2, pp. 65–75, 2017.

[29] B. Rout, B. B. Pati, and S. Panda, "Modified SCA algorithm for SSSC damping controller design in power system," *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, vol. 16, no. 1, pp. 46–63, 2018.

[30] N. Kumar, I. Hussain, B. Singh, and B. K. Panigrahi, "Single sensor-based MPPT of partially shaded PV system for battery charging by using Cauchy and Gaussian sine cosine optimization," *IEEE Transactions on Energy Conversion*, vol. 32, no. 3, pp. 983–992, 2017.

[31] J. Wang, W. Yang, P. Du, and T. Niu, "A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm," *Energy Conversion and Management*, vol. 163, pp. 134–150, 2018.

[32] U. Raut and S. Mishra, "Power distribution network reconfiguration using an improved sine–cosine algorithm-based meta-heuristic search," *Soft Computing for Problem Solving*, pp. 1–13, 2019.

[33] L. Khriddi, N. E. Akkad, H. Satori et al., "Clustering method and sine cosine algorithm for image segmentation," *Evolutionary Intelligence*, vol. 1, pp. 1–14, 2021.

[34] S. Mishra, S. Gupta, and A. Yadav, "Design and application of controller based on sine-cosine algorithm for load frequency control of power system," *Intelligent Systems Design and Applications*, 2020.

[35] X. Xue and J. Chen, "A compact co-firefly algorithm for matching ontologies," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2629–2632, Xiamen, China, September 2019.

[36] X. Xue and J. Chen, "Matching biomedical ontologies through compact differential evolution algorithm," *Systems Science & Control Engineering*, vol. 7, no. 2, pp. 85–89, 2019.

[37] X. Xue, J. Chen, and D. Chen, "Matching biomedical ontologies through compact hybrid evolutionary algorithm," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 1, pp. 110–117, 2019.

[38] S. C. Chu, X. Xue, J. S. Pan et al., "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.

[39] J. F. Chang, J. F. Roddick, J. S. Pan et al., "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, pp. 809–818, 2005.

[40] S. C. Chu, J. F. Roddick, J. S. Pan et al., "Parallel ant colony systems," in *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, pp. 279–284, Knoxville, TE, USA, October 2003.

[41] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[42] H. Wang, S. Rahnamayan, H. Sun, and M. G Omran, "Gaussian bare-bones differential evolution," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 634–647, 2013.

[43] W. J. Cody, "Rational Chebyshev approximations for the error function," *Mathematics of Computation*, vol. 23, no. 107, 631 pages, 1969.

[44] P. C. Song, J. S. Pan, and S. C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Applied Soft Computing*, p. 106443, 2020.

[45] D. Jo, B. Yu, H. Jeon et al., "Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1188–1197, 2018.

[46] W.-K. Lai, T.-H. Kuo, and C.-H. Chen, "Vehicle speed estimation and forecasting methods based on cellular floating vehicle data," *Applied Sciences*, vol. 6, no. 2, p. 47, 2016.

[47] S. Chen, Z. Sun, and B. Bryan, "Traffic monitoring using digital sound field mapping," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 6, pp. 1582–1589, 2001.

[48] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, "Traffic accident prediction using 3-d model-based vehicle tracking," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 677–694, 2004.

[49] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[50] Y. Zhou, Q. Luo, and J. Liu, "Glowworm swarm optimization for dispatching system of public transit vehicles," *Neural Processing Letters*, vol. 40, no. 1, pp. 25–33, 2014.

[51] Z. Hassan and R. Mahmud, "Locating stations of public transportation vehicles for improving transit accessibility," *Transport*, vol. 22, 2007.

[52] B. A. Weigel, F. Southworth, and M. D. Meyer, "Calculators to estimate greenhouse gas emissions from public transit vehicles," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2143, no. 1, pp. 125–133, 2010.

[53] M. Wei, W. Jin, W. Fu et al., "Improved ant colony algorithm for multi-depot bus scheduling problem with route time constraints," in *Proceedings of the 2010 8th World Congress on Intelligent Control and Automation*, pp. 4050–4053, Jinan, China, July 2010.

[54] F. Cevallos and F. Zhao, "A genetic algorithm for bus schedule synchronization," in applications of advanced technology in transportation," in *Proceedings of the Ninth International Conference of ASCE*, Harbin, China, August 2006.

[55] C. Wang, H. Shi, and X. Zuo, "A multi-objective genetic algorithm based approach for dynamical bus vehicles scheduling under traffic congestion," *Swarm and Evolutionary Computation*, vol. 54, p. 100667, 2020.

[56] Z. Yang, S. Zhao, and Q. Zhao, "Research on bus scheduling based on artificial immune algorithm," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, October 2008.

[57] E. Mininno, F. Neri, F. Cupertino et al., "Compact differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 32–54, 2010.

[58] S. Gao, M. Zhou, Y. Wang et al., "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 601–614, 2018.

[59] J. Zhang and S. Member, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[60] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for Differential Evolution," in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 71–78, Cancun, Mexico, June 2013.

[61] R. Tanabe and A. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation*, pp. 1658–1665, Beijing, China, July 2014.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6665539, 12 pages
https://doi.org/10.1155/2021/6665539

Research Article

Modeling and Solution of Vehicle Routing Problem with Grey Time Windows and Multiobjective Constraints

Xiaojian Yuan,[1,2] **Qishan Zhang**,[1] **and Jiaoyan Zeng**[2]

[1]*School of Economics and Management, Fuzhou University, No. 2 Xue Yuan Road, University Town, Fuzhou, Fujian 350116, China*
[2]*School of Technology, Fuzhou University of International Studies and Trade, No. 28, Yuhuan Road, Changle District, Fuzhou, Fujian 350116, China*

Correspondence should be addressed to Xiaojian Yuan; yxj_fz@126.com

Received 3 January 2021; Revised 23 February 2021; Accepted 9 March 2021; Published 26 March 2021

Academic Editor: Cheng Shi

Purpose. In order to study the impact of grey delivery time uncertainty on customer satisfaction and delivery costs, a vehicle routing problem with grey delivery time windows and multiobjective constraints is defined. *Method*. The paper first defines the uncertainty of the delivery vehicle's arrival time to the customer as grey uncertainty and then whitens the grey time windows; at the same time, the customer's hard time windows is expanded into a soft time windows to measure customer satisfaction when the vehicle arrives. *Experiment*. In order to verify the validity of the established model, numerical experiments are carried out in two groups based on the Solomon example, and the solution is solved based on the improved quantum evolution algorithm. *Analysis*. Distribution cost fluctuations and customer satisfaction fluctuations with grey time windows are relatively small; under different satisfaction threshold conditions, the distribution cost is increased gently with the satisfaction threshold. *Conclusion*. The grey delivery time windows have certain advantages in solving the random travel time vehicle routing problem.

1. Introduction

The logistics industry is a composite service industry that integrates transportation, storage, and information industries. It covers a wide range of fields, absorbs a large number of jobs, and plays a major role in promoting production and stimulating consumption. It is an important part of the national economy. How to effectively reduce logistics costs, improve logistics delivery efficiency, provide customers with differentiated services, and improve customer satisfaction are important management issues facing logistics companies. The vehicle routing problem is the core problem in the optimization of logistics distribution, as well as the key problem of smart logistics. Scholars have carried out extensive research on this problem and have achieved fruitful results.

Vehicle routing problem (VRP) is a classic combinational optimization problem with NP. With the rise of modern logistics industry, it has become a key component of intelligent logistics. Since Dantzig and Ramser [1] put forward and studied this problem in depth, scholars have conducted extensive research on the VRP problem and several important variants of this problem and achieved a lot of results. Many of them are based on the actual logistics distribution of scholars who added more constraints so that the research results can be better applied to the actual production.

In VRP variants' problems, as vehicle routing problem with time windows and multiobjective constraints (MOCVRPTW) considers the constraints of customer time windows, customer satisfaction, delivery cost, etc., it is more in line with the "customer first" and in line with the production reality of logistics distribution, which has attracted the attention of academia and industry.

In the actual logistics distribution process, it is necessary to consider not only the uncertainty of vehicle travel time in the distribution process but also how to save the distribution cost to the greatest extent on the basis of meeting the

personalized needs of customers. In this paper, a multi-objective solution method under the constraints of grey time windows is proposed. In terms of the processing of grey time windows, the estimation method of grey time windows under 95% confidence level is given. In terms of satisfaction and distribution cost control, the fuzzy gradient function of customer satisfaction is introduced to set different satisfaction thresholds according to customer classification, and then, the Pareto solution set of different examples is obtained according to the threshold. In order to verify the solvability of the model, based on Solomon's example, this paper extends the customer hard time windows into soft time windows and carries out two groups of numerical experiments. The results show that the fluctuation of distribution cost and customer satisfaction with grey time windows is small. Under different satisfaction threshold conditions, the distribution cost is increased significantly with the increase of satisfaction threshold.

This paper expands the scope of random travel time research and provides decision-making reference for logistics companies. However, this research also has some shortcomings. There is no clear method for customer classification, and the method for setting satisfaction thresholds for different customer categories has not been given. This is also the focus of the author's next research.

The rest of this paper is arranged as follows. After Section 1, related literature will be reviewed in Section 2, and problems are described and modeled in detail in Section 3. Section 4 presents the grey distribution method for generating time windows. Section 5 describes albino customer satisfaction and grey distribution time. Section 6 presents the methods and results of numerical experiment analysis. Finally, the paper makes a summary in Section 7.

2. Literature Review

Scholars generally refer to the problem of uncertain driving time of distribution vehicles as vehicle routing problem with stochastic travel times (VRPSTT). It mainly studies the change of travel time due to traffic condition, road repair, weather, and other factors, and the change of the travel time satisfies some statistical rule of the vehicle routing problem. Laporte et al. [2] established the opportunity constraint and compensation model of the problem. They believed that the punishment for vehicles should be proportional to the length of delay and designed a branch cutting algorithm to solve the problem. Park et al. [3] constructed three different heuristic algorithms to solve this problem based on a modified deterministic VRP solution algorithm. Subsequently, Qiang et al. [4] and Yang et al. [5], respectively, extended and solved the problem on the basis of Laporte research. Zhang et al. [6] defined the uncertainty of travel time as fuzzy travel time, built a mathematical model with fuzzy travel time, and designed a new hybrid genetic algorithm by effectively combining fuzzy logic and fuzzy control method with the genetic algorithm of traditional vehicle routing problem. Li et al. [7] proposed two models of opportunity-constrained programming and stochastic programming with correction for this problem and designed a hybrid algorithm based on tabu search to solve the model. Tao et al. [8] established an opportunity-constrained programming model for random travel time with simultaneous delivery and pickup and designed a decentralized search algorithm solution strategy. Hou et al. [9] also considered the random vehicle routing problem with uncertain demand and uncertain travel time and proposed an improved genetic algorithm with an adaptive mechanism. Tas et al. [10] studied the vehicle routing problem with soft time windows for random travel time and solved the problem based on tabu search algorithm. Li et al. [11] comprehensively considered the total distance and time of vehicles, as well as the equalization of the distance and time of each vehicle, and designed a multiobjective hybrid genetic algorithm combining simulated annealing and genetic algorithm to solve this problem.

In recent years, the rise of e-commerce has led to the vigorous development of the logistics industry, and the research on this issue has maintained a trend of continuous growth. Binart et al. [12] studied this issue with the overall goal of serving the most customers in the shortest time. Miranda and Conceicao [13] focused on the problem that the waiting time after the vehicle arrives at the customer in advance may affect the follow-up distribution arrangement and designed a new method to estimate the time when the vehicle arrives at each customer and the possibility that the vehicle complies with the customer's time windows. Errico et al. [14] proposed a two-stage strategy for this problem and designed an accurate branch cutting algorithm. Gomez et al. [15] relaxed the hypothesis of random travel time distribution of scholars in this question and designed a new solution strategy that could adapt to different travel time distributions. Wang et al. [16] first classified or divided customers into service areas and then designed two heuristic algorithms by introducing intimacy. Miranda et al. [17] studied the multiobjective random traveling time vehicle routing problem with both cost and service level and constructed the multiobjective memetic algorithm and multiobjective iterative local search algorithm. Jianli et al. [18] abstracted the freight distribution routing problem of railway logistics center into the optimization of the batch distribution vehicle routing problem with random travel time and service time, established a stochastic programming model with correction, and designed the corresponding algorithm to solve the model. Guimarans et al. [19] studied the random travel time vehicle routing problem with two-bit packing and designed the path optimization algorithm with Monte Carlo simulation and bias randomness. In order to study the collection of urban garbage, Markovi et al. [20] established a model of simultaneous delivery with random demand and travel time. Hashemi Doulabi et al. [21] solved the VRPSTT problem with synchronous access and service time based on the branch and bound method. Liu [22] studied the side-sharing services and proposed an arc-based vehicle-based integer linear programming model in space-time-state networks, which is solved by Dantzig–Wolfe decomposition.

3. Problem Description and Modeling

3.1. Problem Description. Taking the distribution network between the distribution center and the customer point as the research object, suppose that the distribution center has

some vehicles of the same type with a capacity of Q, and the vehicles provide delivery services for n customers, where customer i (where$i = 1, 2, \ldots, n$) wants that the delivery service of a certain car is obtained within the time period of $[ET_i, LT_i]$. The vehicle departs from each depot, serves several customers in turn, and returns to the original depot after completing the delivery task.

In the actual distribution, due to the uncertainty of traffic and other factors, time t_{ij} spent by the vehicle from customer i to customer j is uncertain. Such uncertainty can be estimated at the upper and lower limits according to experience, which conforms to the characteristics of grey number [23]. Since the customer has uncertainty in the extension of expected time, the paper considers that the delivery time is grey uncertain, and the customer's expected time windows and satisfaction are known in a fuzzy way. Therefore, the paper takes the distribution cost and customer satisfaction as the optimization objectives to carry out modeling and solving and make a reasonable distribution plan.

3.2. Model Assumptions. To facilitate the elaboration and modeling of the problem, this paper makes the following assumptions. (1) All the distribution vehicles have the same models, that is, the load and speed are the same. (2) Determine a number of loops as vehicle driving routes; the total distance of each circuit will not exceed the maximum driving distance of the vehicle; all loops start from the distribution center and return to the vehicle yard; each customer can only be on one loop; each customer on the loop only takes one car's service. (3) The total weight of the customer's goods on each line shall not exceed the vehicle load. (4) The time required for vehicles to enter and exit the distribution center is not taken into account. (5) The service time of all customers shall neither be earlier than the earliest service time tolerated by the customer nor exceed the latest service time tolerated by the customer. Each customer shall have the lowest service level, that is, the needs of the vehicle shall reach the customer's location within the time windows corresponding to the customer's lowest satisfaction. (6) The driving cost per unit routing is fixed, and the goal is to minimize the sum of distribution costs of all vehicles.

3.3. Mathematical Model. For the convenience of illustration and modeling, the sets, parameters, and variables used are defined as follows:

Sets:

(i) V: node set (customer set and distribution center), $V = \{0, 1, 2, \ldots, n\}$ represents the node set of the directed graph, in which 0 represents the distribution center

(ii) V_0: the customer set, $V_0 = \{1, 2, \ldots, n\}$

(iii) K: the vehicle set, $K = \{1, 2, \ldots, k\}$

Parameters:

(i) Q: the rated load of the vehicle

(ii) c_d: the cost of starting a vehicle

(iii) c_t: the driving cost of a single vehicle per unit distance

(iv) d_{ij}: Euclidean distance between customer i and j, $i, j \in V_0, i \neq j$

(v) d_{0i}: the distance between the distribution center and the customer i, and $d_{0i} = d_{i0}$, $i \in V_0$

(vi) D_i: the total weight of all the goods of the customer i, $i \in V_0$

(vii) $[ET_i, LT_i]$: the expected service time windows of customer i; ET_i is the earliest service time expected by the customer and LT_i is the latest service time expected by the customer, $i \in V_0$

(viii) $[ET_0, LT_0]$: opening time of the distribution center

(xi) EET_i: the earliest service time tolerated by the customer i, $i \in V_0$

(x) ELT_i: the latest service time tolerated by the customer i, $i \in V_0$

(xi) st_i: the time required for the vehicle to serve the customer i, $i \in V_0$

(xii) t_{ij}: the time taken by the vehicle from the customer i to j, $i, j \in V_0, i \neq j$

(xiii) t_{ij}^R: the time taken by the vehicle from the customer i to j at a constant speed state, $i, j \in V_0, i \neq j$

(xiv) t_i: the time the vehicle arrives at the customer i, $i \in V_0$

(xv) $U_i(t_i)$: the satisfaction of customer i at time t_i in the car

(xvi) λ: the parameters used to weigh dispatch costs against running costs, $\lambda \in [0, 1]$

Decision variables:

(i) x_{ijk}: the driving variable of vehicle k ($k \in K$), and $x_{ijk} \in \{0, 1\}$. If the vehicle k travels from point i to point j, then $x_{ijk} = 1$, otherwise $x_{ijk} = 1$.

(ii) L_{0k}: the weight of the goods carried by the vehicle k when it leaves the distribution center.

(iii) t_{ik}: the time when the vehicle k starts to serve the customer i; if the vehicle k does not serve the customer i, then $t_{ik} = 0$, $i \in V_0, k \in K$.

Based on the above variables, the MIP (mixed integer programming) model of the problem is given:

$$\text{Min} \quad Z = \lambda \sum_{j \in V_0} \sum_{k \in K} c_d x_{0jk} + (1 - \lambda) \sum_{i \in V_0} \sum_{j \in V_0} \sum_{k \in K} c_t d_{ijk} x_{ijk}, \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in V_0} \sum_{k \in K} x_{ijk} = 1, \tag{2}$$

$$\sum_{i \in V_0} x_{ihk} = \sum_{j \in V_0} x_{hjk}, \quad h \in V_0, \tag{3}$$

$$\sum_{j \in V_0} x_{0jk} = \sum_{i \in V_0} x_{i0k}, \tag{4}$$

$$L_{0k} = \sum_{i \in V_0} \sum_{j \in V_0} q_i x_{ijk}, \tag{5}$$

$$L_{0k} \leq Q, \tag{6}$$

$$EET_i \leq t_{ik} \leq ELT_i, \tag{7}$$

$$x_{ijk} \in \{0, 1\}, \quad i \in V_0, j \in V_0, k \in K, \tag{8}$$

where equation (1) is the objective function of the problem. Constraint (2) means that each customer can only be served by one car, that is, the service cannot be split. Constraint (3) avoids the formation of a loop that does not include the distribution center. Constraint (4) ensures that the initial starting point and final point of the vehicle are distribution centers, Constraints (5) and (6) are vehicle load, that is, the total weight of the goods carried by the vehicle shall not be greater than the vehicle load. Constraint (7) is the time windows constraint.

Different logistics enterprises can change the value of λ according to their own conditions, that is, change the value of $(\lambda/1 - \lambda)$. Then, the model is applicable to the minimum priority of vehicles, the minimum priority of total distance, and other problems in between. For example, if the logistics company has enough vehicles and drivers, then the company does not need to care too much about the startup cost of the vehicle. The value of λ can be appropriately reduced, or even $\lambda = 0$. At this time, the vehicle optimization goal is the shortest total delivery distance and no longer considers vehicle startup costs. If the logistics company has insufficient delivery vehicles or insufficient drivers, then it is necessary to consider a larger value of λ, then appropriately increase the value of c_d, then the minimum number of vehicles used becomes the first optimization goal, and the total delivery distance becomes the second optimize the target.

4. Grey Delivery Time Windows

Under normal circumstances, the time to reach the destination is often different due to road traffic conditions or nonuniform speed of the vehicle, resulting in uncertainty in the time for the vehicle to reach the customer, but the range of arrival time can be estimated from historical data. Therefore, this paper assumes that the delivery time is grey, and the travel budget time equation in [24] generates vehicle delivery grey time. The specific method is as follows.

The expected travel time from node i to node j is expressed as the sum of time and error time under uniform speed and can be expressed as

$$t_{ij}^h = t_{ij}^R \pm S_{ij}. \tag{9}$$

Then, the estimated latest arrival time t_{ij}^u is the pessimistic value $t_{\inf}$ of the delivery time when the confidence level is $1 - \alpha$ (generally, the confidence is 95%, at this time $\alpha = 0.05$), namely,

$$t_{\inf} = \inf\left\{t_{ij}^h \middle| \Pr\left\{t_{ij} \leq t_{ij}^h\right\} \geq \alpha\right\} = t_{ij}^R + S_{ij}. \tag{10}$$

The time from node i to node j is affected by many independent random factors. According to the central limit theorem, the sum of the superimposed effects of many small independent factors obeys or approximately obeys a normal distribution [18, 25]. It is assumed that the actual arrival time t_{ij} obeys a normal distribution, that is, $t_{ij} \sim N(t_{ij}^R, \sigma_{ij}^2)$, where σ_{ij}^2 represents the variance of the travel time of the vehicle on the road section ij.

Therefore, equation (10) can also be expressed as

$$\begin{aligned} &\min \quad t_{ij}^h \\ &\text{s.t} \quad \int_{-\infty}^{t_{ij}^h} \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\left(\left(x - t_{ij}^R\right)^2 / 2\sigma_{ij}^2\right)} \mathrm{d}x \geq \alpha. \end{aligned} \tag{11}$$

First standardize equation (11) and then solve it to get $S_{ij} = \sigma_{ij}\Phi^{-1}(\alpha)$. Since the minimum target value is required, equation (11) can be obtained:

$$S_{ij} = \sigma_{ij}\Phi^{-1}(\alpha). \tag{12}$$

Then, the latest arrival time of the vehicle can be expressed as

$$t_{ij}^u = t_{ij}^R + \sigma_{ij}\Phi^{-1}(\alpha). \tag{13}$$

In the same way, the expression for the earliest arrival time can be obtained:

$$t_{ij}^l = t_{ij}^R - \sigma_{ij}\Phi^{-1}(\alpha). \tag{14}$$

In summary, the value range of the grey number $\otimes t_{ij}$ from node i to node j can be obtained, that is, the grey delivery time windows:

$$\left[t_{ij}^R - \sigma_{ij}\Phi^{-1}(\alpha), t_{ij}^R + \sigma_{ij}\Phi^{-1}(\alpha)\right]. \tag{15}$$

5. Customer Satisfaction

5.1. Fuzzify Customer Time Windows. In the delivery process, customer satisfaction is mainly reflected in whether the vehicle can arrive on time. If the vehicle arrives within the customer's expected time, the customer's satisfaction is the highest, and the delayed arrival will cause a decrease in customer satisfaction. In order to reflect the customer's expected time more accurately, combined with time elements and the actual needs of the customer, the customer's expected time windows is first blurred, and then, the fuzzy gradient function is used to express the customer's satisfaction. The fuzzification rules refer to the method of Wang et al. [26]; in the paper by Wang, if the time windows is extended too much, such as extending 100 to both sides, the average customer satisfaction will be higher. At this time, the problem is close to the problem of no time windows requirement, and it is not easy to distinguish various routes. If the expansion is too small, such as 30 for each expansion, the satisfaction degree will be too large, and the average satisfaction degree will be too strongly correlated with the time windows. At this time, the problem is close to the hard time

windows requirement, and the expected effect is also not achieved.

Based on the research of Wang et al., extend the time windows by 50 on both sides, and the time windows is fuzzified according to equations (16) and (17):

$$EET_i = \begin{cases} ET_i - 50, & ET_i \geq 50, \\ 0, & ET_i < 50, \end{cases} \tag{16}$$

$$ELT_i = LT_i + 50. \tag{17}$$

Assuming that the time the vehicle arrives at the customer i is t_i, the membership function expression used in this paper to deal with the customer time windows is

$$U_i(t_i) = \begin{cases} \dfrac{\otimes t_i - EET_i}{ET_i - EET_i}, & \otimes t_i \in [EET_i, ET_i), \\ 1, & \otimes t_i \in [ET_i, LT_i], \\ \dfrac{ELT_i - \otimes t_i}{ELT_i - LT_i}, & \otimes t_i \in (LT_i, ELT_i], \\ 0, & \otimes t_i \notin [EET_i, ELT_i]. \end{cases} \tag{18}$$

The meaning of equation (18) is that if the customer arrives before the earliest time EET that the customer can tolerate or arrives after the latest time ELT that can be tolerated, the customer is no longer in working hours and cannot deliver the goods, so customer satisfaction is 0. If after the earliest tolerated time EET but before the normal work start time ET or after the normal working time LT but before the latest tolerated time ELT, customer satisfaction can hand over the goods because it is not part of the normal working hours and satisfaction is not high, but the satisfaction level is higher when the arrival time is about close to normal working hours. If the vehicle arrives during normal working hours [ET, LT], the customer is most satisfied with 100% satisfaction (here, marked as 1).

In order to express it more intuitively, use the polyline expression (18) in the rectangular coordinate system, as shown in Figure 1. It can be seen from Figure 1 that

(1) If $\otimes t_i$ is before the earliest time, the customer can tolerate, which means $\otimes t_i \leq EET_i$; then, the waiting time windows opens, and the customer satisfaction is 0

(2) If $\otimes t_i$ exceeds the time windows expected by the customer, but is after the earliest tolerable time, $\otimes t_i \in [EET_i, ET_i)$, the customer satisfaction value is between [0, 1)

(3) If $\otimes t_i$ is within the service time windows $[ET_i, LT_i]$, then the customer satisfaction is 1

(4) If $\otimes t_i$ exceeds the time windows expected by the customer, but is before the latest tolerable time, $\otimes t_i \in (LT_i, ELT_i]$; then, the customer satisfaction is [0, 1)

(5) If $\otimes t_i$ exceeds ELT_i, which means $\otimes t_i > ELT_i$, then the customer satisfaction is 0

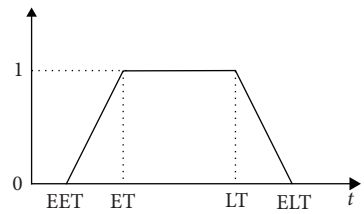


Figure 1: Fuzzy gradient function of customer satisfaction.

5.2. Satisfaction Threshold. In the actual distribution, the demands of different customer satisfaction are generated according to different products or important levels of customers. In order to save cost and provide personalized service, customers are classified and the minimum satisfaction threshold is set. As shown in Figure 2, for a certain customer j, the minimum satisfaction threshold is l_j. Then, corresponding to this threshold, the time interval $[t_{js}, t_{je}]$ for which the vehicle can serve the customer is the part of the broken line in bold, as shown in Figure 2, which is the part above the line $y = l_j$ (including the equal).

5.3. The Matching Problem of Grey Time Windows and Customer Satisfaction Threshold. According to equation (13), under a given confidence level 1-α, the latest time for the vehicle to reach customer i is $t_{ij}^R + \sigma_{ij}\Phi^{-1}(\alpha)$, so as long as $t_{ij}^R + \sigma_{ij}\Phi^{-1}(\alpha) \leq t_{je}$, and the vehicle can decide to depart from customer i to customer j; otherwise, there will be the risk of not being able to deliver even if it reaches customer j.

5.4. Whitening of Grey Delivery Time. Although 5.3 gives the conditions for the selection of the delivery vehicle route, the subsequent calculation requires the time for the vehicle to actually reach the customer j, so the grey number $\otimes t_{ij}$ needs to be whitened [27].

For the whitening weight function $g(x)$ of $\otimes t_{ij}$ (assuming continuous and nonnegative), perform the following function transformation to get its whitening value:

(1) According to the definition of probability density function, $f(x)$ is the probability density function of random variable x, and the expression is

$$f(x) = \frac{g(x)}{\int_a^b g(z)\mathrm{d}z}. \tag{19}$$

(2) Then, the distribution function $F(x)$ can be obtained:

$$F(x) = \frac{\int_a^x g(z)\mathrm{d}z}{\int_a^b g(z)\mathrm{d}z}, \quad F(x) \in [0, 1]. \tag{20}$$

(3) Therefore, as long as the random variable u that obeys [0, 1] is generated, the random whitening value $\bar{\otimes}$ of $\otimes t_{ij}$ can be obtained by calculating $F^{-1}(u)$.

Therefore, the calculation method for the specific time t_j of the delivery vehicle arriving at the customer j via the customer i is as follows (here, assuming that the vehicle

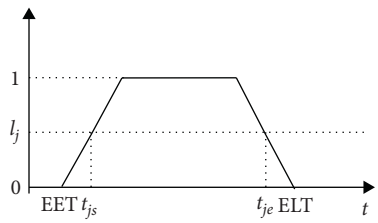


Figure 2: Delivery time corresponding to a specific satisfaction threshold.

arrives at i, that is, it will start to provide delivery services for i, there is no waiting time; in actual calculations, if there is a waiting, you need to add waiting time):

$$t_j = t_i + st_i + \bar{\otimes} t_{ij}. \tag{21}$$

5.5. Satisfaction Calculation. The pseudocode for customer satisfaction calculation is as follows:

(i) Step 1: set the satisfaction threshold

(ii) Step 2: calculate the number of customers currently served by the vehicle

(iii) Step 3: enumerate customers on the current vehicle path

  (1) Step 3.1: according to (21), calculate the time for the vehicle to leave the current customer and arrive at the next customer

  (2) Step 3.2: refer to equation (18) and the satisfaction threshold to calculate time $t_{j\text{start}}$ to start serving customer j

  (3) Step 3.3: bring the service start time calculated in Step 3.2 into equation (18) to calculate the satisfaction of customer j who is receiving service

  (4) Step 3.4: calculate the departure time according to the equation $t_{j\text{start}} + ST_j$, and jump out of the current cycle if all customers to be served by the vehicle are serviced

(iv) Step 4: calculate the total satisfaction of the current vehicle

6. Experiment and Analysis

In order to solve the model, the author expands the hard time windows into a soft time windows in Solomon's example and conducts numerical experiments in two groups based on the improved quantum evolution algorithm [28]. A new idea is provided for solving the problem of serious loss of effective information in the quantum evolutionary algorithm by defining the quantum cell body.

There are 3 types of Solomon's calculation examples: Rdp, Cdp, and Rcdp. The geographic location of customers in the Rdp calculation example meets a random distribution, the geographic location of customers in the Cdp calculation example meets the clustering distribution, and the geographic location of the customer in the Rcdp calculation example is generated by a mixture of random distribution and cluster distribution. Except for the 9 small- and medium-sized customers, the customer size of the remaining 56 examples is 100. Each type of example can be divided into two categories: the first type of vehicle has a short planning time and weak installation capacity. The second type of vehicles has a long planning time and strong installation capacity; among them, Rdp1, Cdp1, and Rcdp1 have a short planning time and a low vehicle loading capacity (both 200), so the customers that each delivery vehicle can serve are relatively limited (about 5–10 customers), while the planning time for Rdp2, Cdp2, and Rcdp2 is longer, the loading capacity of the vehicle is larger (1000, 700, and 1000, respectively), and each delivery vehicle can serve more customers (some routes can exceed 30).

Considering the generality and time economy, take one from the middle position for each small category in the 56 examples of large-scale customers for numerical experiments. The data characteristics of the six calculation examples are shown in Table 1. The table lists the number, number of customers, total number of items, and number of available vehicles for each test case in Solomon's example.

6.1. Service Quality Analysis under the Constraint of Grey Time Windows. Taking into account that most scholars' research takes the shortest distance as the optimization goal, in order to facilitate comparison and analysis, in the experiment, this paper also assumes that the vehicle startup cost is 0, and the objective function is $\lambda = 0$ in equation (1). The unit journey cost of the vehicle is $c_t = 1$, the quantum evolution algorithm population is 30, and the maximum number of iterations is set to 100.

In order to verify the difference in service quality when the delivery time windows is grey and the delivery time is fixed, 10 experiments were carried out using 6 selected examples. Name the experiment with a grey delivery time windows as Test1, and name the experiment with the vehicle driving at a constant speed, that is, with a fixed delivery time, as Test2. The customer satisfaction and distribution cost information obtained are shown in Table 2.

As can be seen from Table 2, in terms of customer satisfaction, the average variance of the five calculations for customer satisfaction in Test1 is 0.0002514 and Test2 is 0.001427. In terms of distribution cost, the average variance of the five calculations for the distribution cost in Test1 is 38.71 and that of Test2 is 814.58. The above data shows that, under the grey delivery time windows, the volatility of customer satisfaction and delivery costs is small, and the overall effect of the model is better.

In order to compare the effects of two time processing methods of grey delivery time and random delivery time on vehicle driving distance and customer satisfaction, set α in the study by Qiang et al. [4] to 0.9, and a set of experiments is designed using the above method, named Test3, and the results show that the two methods have no difference in the impact of delivery distance. In terms of customer satisfaction, the variance of customer satisfaction of Test3 is 0.0003673. The results show that the grey time windows method proposed in this paper has certain advantages.

Table 1: The characteristics of the test cases selected in this paper.

	Number	Number of customers	Vehicle load	Available vehicles
1	C105	100	200	25
2	C205	100	700	25
3	R105	100	200	25
4	R205	100	1000	25
5	RC105	100	200	25
6	RC205	100	1000	25

Table 2: Statistics of calculation results.

Example number	Experiment name	Customer satisfaction			Distribution cost		
		Maximum	Minimum	Variance	Maximum	Minimum	Variance
C105	Test1	0.98	0.97	0.0000593	369.74	363.25	9.38
	Test2	0.98	0.90	0.0014682	372.32	363.25	18.29
C205	Test1	0.90	0.88	0.0000457	361.80	361.41	0.04
	Test2	0.89	0.86	0.0001539	380.49	361.41	77.41
R105	Test1	0.81	0.77	0.0004081	843.32	828.81	35.54
	Test2	0.82	0.72	0.0014266	908.23	814.62	1620.04
R205	Test1	0.89	0.84	0.0004327	789.13	771.72	59.07
	Test2	0.91	0.83	0.0011436	796.22	682.84	2353.54
RC105	Test1	0.85	0.82	0.0001368	660.71	635.92	106.68
	Test2	0.87	0.82	0.0000249	736.14	681.30	645.32
RC205	Test1	0.88	0.83	0.0004262	647.19	637.12	21.57
	Test2	0.91	0.76	0.0038700	664.47	632.77	172.86

Table 3: Distribution costs and average satisfaction under different satisfaction thresholds.

	$S_{\lim}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
C105	C_f	363.25	363.25	369.74	372.32	392.63	401.88	406.18	412.96	417.39	445.12
	S_f	0.67	0.75	0.77	0.78	0.81	0.94	0.96	0.97	0.99	1.00
C205	C_f	141.54	245.98	249.35	250.15	269.00	326.52	380.49	443.23	461.85	479.11
	S_f	0.42	0.62	0.63	0.69	0.77	0.80	0.84	0.87	0.90	1.00
R105	C_f	609.70	646.43	654.70	692.30	727.23	750.84	796.38	800.95	814.62	905.56
	S_f	0.47	0.55	0.57	0.66	0.69	0.74	0.80	0.82	0.88	1.00
R205	C_f	745.62	771.72	772.37	774.22	789.13	792.56	796.22	824.96	842.06	864.15
	S_f	0.55	0.73	0.74	0.76	0.80	0.84	0.87	0.93	0.97	1.00
RC105	C_f	558.88	597.58	606.66	611.61	620.05	663.01	736.14	762.89	784.18	805.57
	S_f	0.63	0.65	0.69	0.74	0.80	0.84	0.88	0.89	0.90	1.00
RC205	C_f	621.15	627.88	632.60	635.74	637.57	640.39	647.19	664.47	677.83	681.73
	S_f	0.59	0.61	0.71	0.72	0.72	0.83	0.86	0.88	0.92	1.00

6.2. Experiment and Analysis under Different Satisfaction Thresholds. In order to provide customers with more personalized services while saving business costs, different satisfaction thresholds (the minimum satisfaction of a single customer) can be set for different customers when providing delivery services to customers.

Taking equation (1) as the objective function, without considering the vehicle startup cost, the experimental environment and experimental parameters are the same as 6.1, the customer satisfaction threshold $S_{\lim}$ is from 0.1 to 0.9, the step size is 0.1, and the maximum number of iterations for each threshold is set to 100. Calculate the grey distribution time windows, the distribution cost C_f of different threshold constraints, and the average customer satisfaction S_f. The results are shown in Table 3.

In order to facilitate understanding and more intuitive performance of the data in Table 3, using Figures 3–8, respectively, the five examples of distribution costs and average under different satisfaction thresholds are shown.

In the six figures, the 10 small circles on the right of each picture correspond to the 10 sets of data in Table 3. The last small blue glowing circle corresponds to the data with a satisfaction degree of 1. When the customer satisfaction threshold is set to 1, it means that the vehicle must arrive within the time windows $[ET, LT]$ expected by the customer.

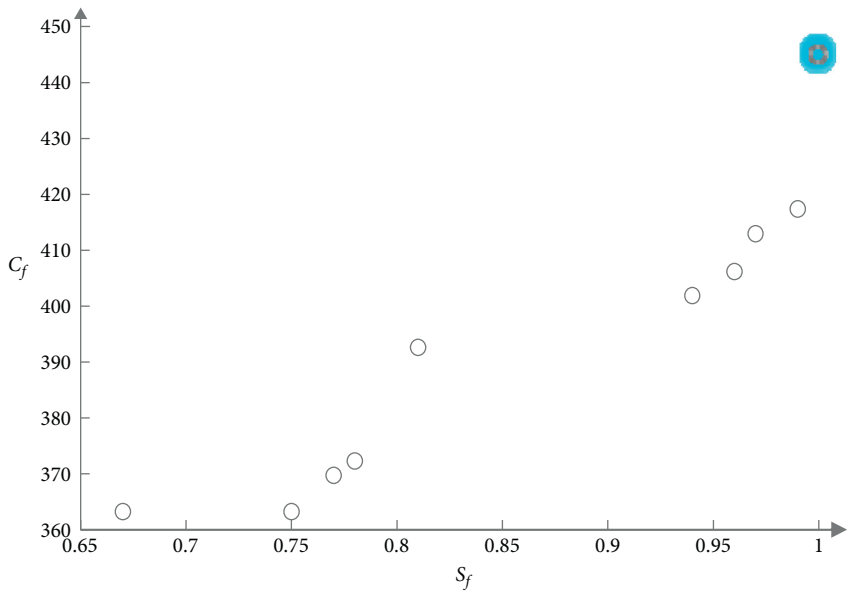


Figure 3: Noninferior solution set of example C105.

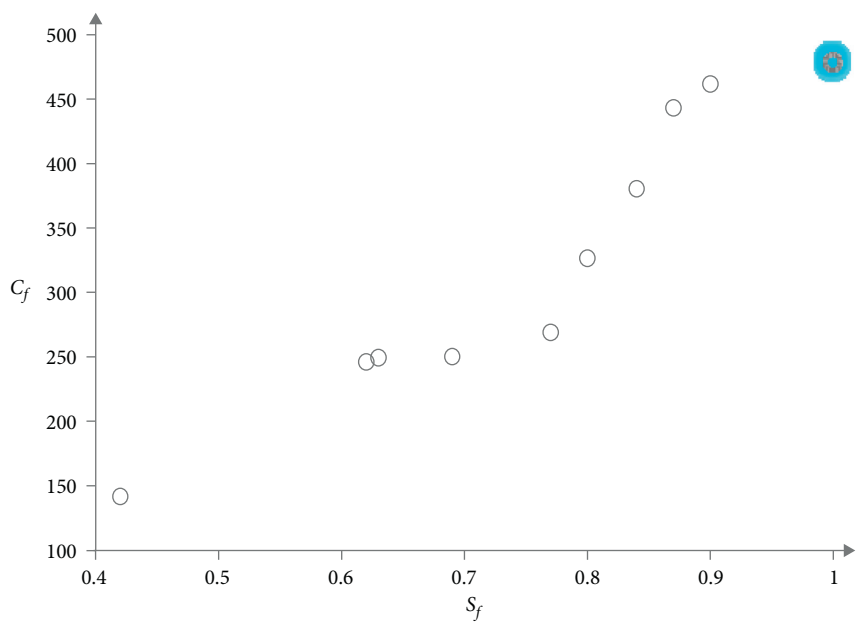


Figure 4: Noninferior solution set of example C205.

At this time, the problem is transformed into a hard time windows constrained vehicle routing optimization problem (CVRPTW).

This experiment shows that taking into account customer satisfaction, the distribution cost is significantly increased; the distribution cost reaches the highest when the customer satisfaction is 1, that is, under the hard time windows; the distribution cost increases as the satisfaction threshold increases, and the distribution cost gradually increases. In actual production, distribution companies can classify customers and provide personalized distribution services for different customers, so as to achieve the goal of

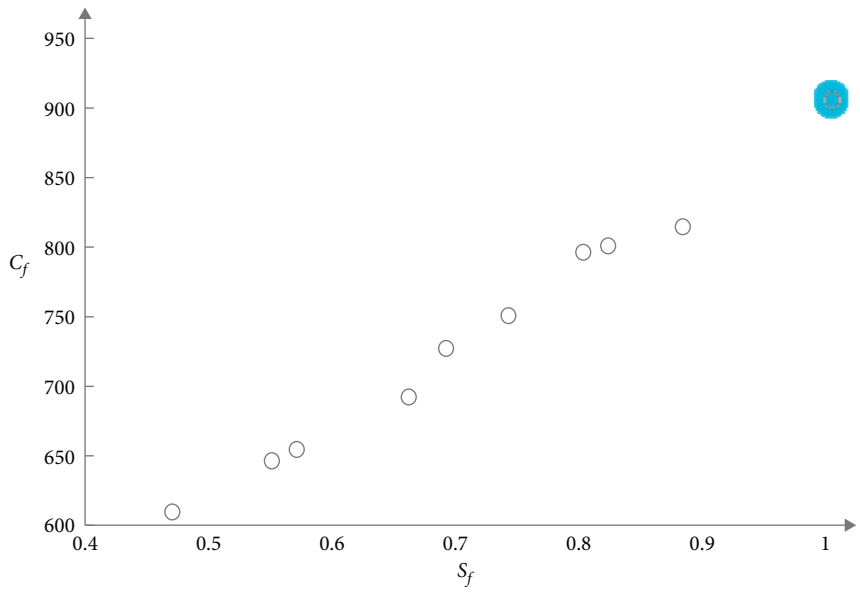


Figure 5: Noninferior solution set of example R105.

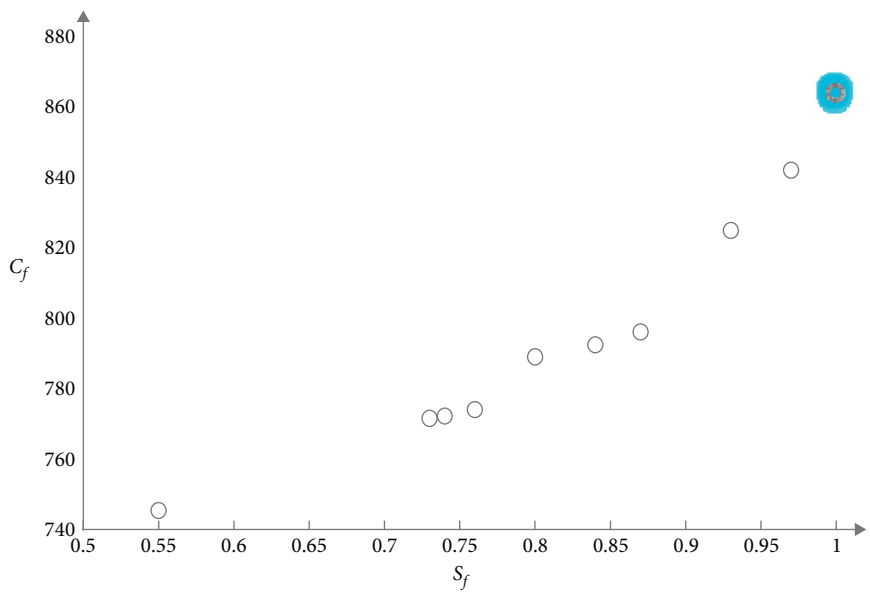


Figure 6: Noninferior solution set of example R205.

reducing business operating costs without affecting service quality.

In addition, it can be seen from Figures 3 to 8 that the distribution cost gradually increases with the relatively smooth customer satisfaction, indicating that the grey time windows has certain advantages in solving this kind of random travel time vehicle routing problem.

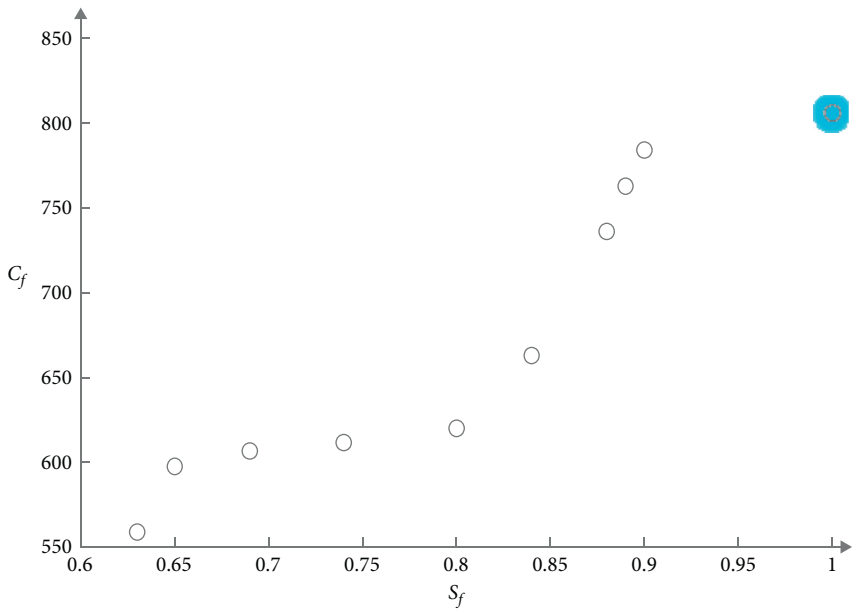


Figure 7: Noninferior solution set of example RC105.

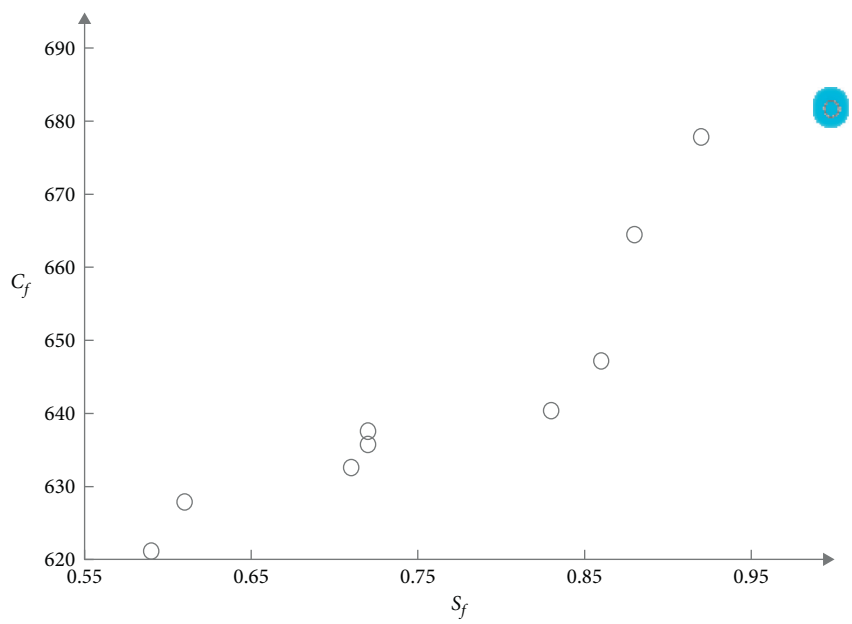


Figure 8: Noninferior solution set of example RC205.

7. Conclusion

The paper abstracts the common problem of uncertain distribution of road conditions into the problem of uncertain distribution time. Combining the grey time windows of delivery with the customer time windows, the choice of vehicle delivery route is decided and customer satisfaction is determined, new grey time windows and multiobjective vehicle routing problem are constructed, and the construction method and whitening of the grey time windows method are given. Finally, based on the improved quantum evolution algorithm, the author designed two sets of numerical experiments on the modified Solomon's examples. Both sets of experimental results show that the grey time windows has certain advantages in solving the dynamic travel time of the vehicle path and can provide a reference for logistics distribution enterprises.

The grey delivery time windows proposed in this paper expands the research scope of the vehicle routing problem. Since no actual logistics data was used for experiments, this study also has limitations. Simulation experiments based on the operating data of large logistics companies are also the direction of the authors' future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the following funds: Major Educational and Teaching Reform Research Project of Undergraduate Universities in Fujian Province (FBJG20190284) and Educational and Scientific Research Support Project for Young and Middle-aged Teachers of Fujian Provincial Department of Education (JT180679).

Supplementary Materials

Supplementary 1. The distribution plan corresponding to Table 2: the table lists the distribution plans corresponding to the maximum and minimum values in the five experiments in Table 2 in the paper, where 0 represents the starting point of distribution, and each vehicle starts from 0 to end. *Supplementary 2.* The distribution plan corresponding to Table 3: the table lists the distribution schemes under each satisfaction threshold corresponding to Table 3 in the paper, where 0 represents the starting point of distribution, and each vehicle starts from 0 to end. (*Supplementary Materials*)

References

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] G. Laporte, F. Louveaux, and H. Mercure, "The vehicle routing problem with stochastic travel times," *Transportation Science*, vol. 26, no. 3, p. 161, 1992.

[3] Y.-B. Park and S.-H. Song, "Vehicle scheduling problems with time-varying speed," *Computers and Industrial Engineering*, vol. 33, no. 3-4, pp. 853–856, 1997.

[4] G. Qiang and X. Binglei, "Model and algorithm of vehicle routing problem with stochastic travel time," *Journal of Systems Engineering*, vol. 18, no. 3, pp. 244–247, 2003.

[5] Z. Yang, H. Qing, B. U. Xiang-zhi, and A. Gutierrez, "Model and algorithm for on-line vehicle routing problem with stochastic travel time," *Journal of Industrial Engineering and Engineering Management*, vol. 20, no. 3, pp. 82–84, 2006.

[6] J. Zhang and J. Li, "A hybrid genetic algorithm to the vehicle routing problem with fuzzy traveling time," *Journal of Industrial Engineering/Engineering Management*, vol. 20, no. 4, pp. 13–16, 2006.

[7] X. Li and P. Tian, "Vehicle routing problems with time windows and stochastic times:Models & algorithm," *System Engineering Theory and Practice*, vol. 29, no. 8, pp. 81–90, 2009.

[8] Z. Tao, Y. Chuo-ya, L. Lan, S. Zhi-Fang, and Z. Yue-Jie, "Method for the stochastic traveling time VRPSPD problem," *System Engineering Theory and Practice*, vol. 31, no. 10, pp. 1912–1920, 2011.

[9] L. Hou, H. Zhou, and C. Liang, "Vehicle routing problem with uncertain demand and travel time," *Computer Integrated Manufacturing Systems*, vol. 17, no. 1, pp. 101–108, 2011.

[10] D. Taş, N. Dellaert, T. van Woensel, and T. de Kok, "Vehicle routing problem with stochastic travel times including soft time windows and service costs," *Computers and Operations Research*, vol. 40, no. 1, pp. 214–224, 2013.

[11] F. Li and Y. Wei, "Hybrid genetic algorithm for capacitated vehicle routing problem with stochastic travel time," *Journal of Industrial Engineering and Engineering Management*, vol. 23, no. 6, pp. 819–825, 2014.

[12] S. Binart, P. Dejax, M. Gendreau, and F. Semet, "A 2-stage method for a field service routing problem with stochastic travel and service times," *Computers and Operations Research*, vol. 65, pp. 64–75, 2016.

[13] D. M. Miranda and S. V. Conceição, "The vehicle routing problem with hard time windows and stochastic travel and service time," *Expert Systems with Applications*, vol. 64, no. 64, pp. 104–116, 2016.

[14] F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau, "A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times," *European Journal of Operational Research*, vol. 249, no. 1, pp. 55–66, 2016.

[15] A. Gómez, R. Mariño, R. Akhavan-Tabatabaei, A. L. Medaglia, and J. E. Mendoza, "On modeling stochastic travel and service times in vehicle routing," *Transportation Science*, vol. 50, no. 2, pp. 627–641, 2016.

[16] Z. Wang and W.-H. Lin, "Incorporating travel time uncertainty into the design of service regions for delivery/pickup problems with time windows," *Expert Systems with Applications*, vol. 72, pp. 207–220, 2017.

[17] D. M. Miranda, J. Branke, and S. V. Conceição, "Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time," *Applied Soft Computing*, vol. 70, pp. 66–79, 2018.

[18] S. Jianli and Z. Jin, "Optimization on simultaneous pick-up and delivery vehicle routing problem with split delivery and stochastic travel and service time," *Control and Decision*, vol. 33, no. 4, pp. 657–670, 2018.

[19] D. Guimarans, O. Dominguez, J. Panadero, and A. A. Juan, "A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times," *Simulation Modelling Practice and Theory*, vol. 89, pp. 1–14, 2018.

[20] D. Marković, G. Petrović, Ž. Ćojbašić, and D. Marinković, "A metaheuristic approach to the waste collection vehicle routing problem with stochastic demands and travel times," *Acta Polytechnica Hungarica*, vol. 16, no. 7, pp. 45–60, 2019.

[21] H. Hashemi Doulabi, G. Pesant, and L.-M. Rousseau, "Vehicle routing problems with synchronized visits and stochastic travel and service times: applications in healthcare," *Transportation Science*, vol. 54, no. 4, pp. 1053–1072, 2020.

[22] J. Liu, P. Mirchandani, and X. Zhou, "Integrated vehicle assignment and routing for system-optimal shared mobility planning with endogenous road congestion," *Transportation Research Part C: Emerging Technologies*, vol. 117, Article ID 102675, 2020.

[23] S. Liu, *Grey System Theory and its Application*, Science Press, China, 5th edition, 2010.

[24] H. K. Lo, X. W. Luo, and B. W. Y. Siu, "Degradable transport network: travel time budget of travelers with heterogeneous risk aversion," *Transportation Research Part B: Methodological*, vol. 40, no. 9, pp. 792–806, 2006.

[25] M. A. Hui-Ru, F. Zhao, J. Li-Min, and Z. Xing-Chen, "Travel salesman problem and solving algorithm based on stochastic

chance-constrained programming model," *Journal of Chang'an University(Natural Science Edition)*, vol. 35, no. S1, pp. 179–183, 2015.

[26] X. Wang, J. Ruan, K. Zhang, and C. Ma, "Study on combinational disruption management for vehicle routing problem with fuzzy time windows," *Journal of Management Sciences in China*, vol. 14, no. 6, pp. 2–15, 2011.

[27] Y. Xiaojian, Z. Qishan, and L. Hong, "Solving MDVRP with grey delivery time based on improved quantum evolutionary algorithm," *The Journal of Grey System*, vol. 32, no. 3, pp. 110–123, 2020.

[28] X. Yuan, Q. Zhang, L. Wu, and Y. Jiang, "Solving VRPSPDTW problem based on improved quantum algorithm," *Journal of Fuzhou University (Natural Science Edition)*, vol. 48, no. 5, pp. 1–7, 2020.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6624129, 22 pages
https://doi.org/10.1155/2021/6624129

Research Article

An Intelligent Framework for Analyzing the Feasible Modes of Transportation in Metropolitan Cities: A Hybrid Multicriteria Approach

Praveen Ranjan Srivastava,[1] **Zuopeng (Justin) Zhang**,[2] **Prajwal Eachempati**,[1] **and Hongbo Lyu**[3]

[1]*Indian Institute of Management (IIM), Rohtak, India*
[2]*Department of Management, Coggin College of Business, University of North Florida, Jacksonville, FL, USA*
[3]*Zhejiang Wanli University, Ningbo, China*

Correspondence should be addressed to Hongbo Lyu; lvhongbo@zwu.edu.cn

Received 12 December 2020; Revised 24 January 2021; Accepted 16 February 2021; Published 1 March 2021

Academic Editor: Chi-Hua Chen

The paper aims to build a hybrid personalized multicriteria model in the Indian transportation industry to identify the most feasible transport mode suitable for commuters' customized preferences. A hybrid multicriterion model, i.e., Fuzzy Analytical Hierarchy Process (AHP), was used to compute the criteria weights, which were subsequently analyzed by three approaches, namely, Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), Fuzzy TOPSIS, Evaluation Based on Distance from Average Solution (EDA), and Interpretive Ranking Process (IRP). The case of an Indian metropolitan city, Hyderabad, is taken to illustrate the proposed approach. The paper highlights the following transport modes: metropolitan train (unconventional mode) and conventional modes such as the car, public bus transport, and bikes for Hyderabad. Furthermore, sensitivity analysis is performed to identify the consistency in ranking with variation in weights, and the Ensemble Ranking and transportation experts validate the rankings.

1. Introduction

India's transportation industry has been growing steadily at a Cumulative Annual Growth Rate (CAGR) of 5.9% and is majorly dominated by roadways. India has a road network of 5.23 million kilometers and is expected to grow at a CAGR of 7% in the next five years [1]. It is also reported by the Mordor Intelligence Report [1] that more than 50% of the freight and 90% of the passengers are found to commute by road. This may contribute to a manifold increase in road traffic, and there is a need for more effective traffic management and vehicle routing. Passengers, especially across such traffic-intensive routes, find it challenging to travel by conventional transport modes like public buses within the city. The development of cars due to the proliferation of automobile technology is increasing by leaps and bounds. However, the notion of cars is not affordable and accessible to many, which prompts the need to search for more available modes of transport, which are cost-effective, environmentally less damaging, safe, and more affordable. In the light of this scenario, the metro train alternative is explored and analyzed for viability for transporting passengers across long distances within the confines of a city with minimal damage to the environment and nominal ticket rates.

Existing decision-making systems do not consider the customized preferences of the passengers and the transport planners to provide an automated decision regarding the vehicle mode is feasible, which is essential for a developing country and a price-sensitive market like India. Moreover, any country's Central government accords paramount importance to the safety and security of the conventional modes of transportation apart from analyzing the economic

feasibility and the environmental sustainability. Some factors like political consequences, cost-effectiveness, and the impact on the environment [2] are prioritized while choosing the best-suited vehicle mode. The simulation model develops various permutations and combinations to examine the possibilities. It is envisaged that the transport authorities and the planners would benefit from the study to meet the growing demand from passengers.

In this context, there is a need to choose a vehicle mode that satisfies all the stakeholders (users, operators, planners, and policymakers). The choice must also rationalize all the stakeholders' conflicting perspectives for which the hybrid multicriteria model is adopted in this paper. The PESTLE framework (Political, Economic, Social, Technological, Legal, and Ecological) is adopted in this paper to identify the most suitable transport alternative with due consideration for all dimensions. This framework is appropriate since different stakeholders consider different perspectives. Users consider mainly the economic and social perspectives to decide on the choice of the vehicle mode. If the choice is studied ony from a user perspective, other (political, technological, and legel) considerations may not be factored in, thus resulting in a mopic viewpoint. However, in the context of a metropolitan city and for a problem statement like the vehicle mode choice, there is a need to address all the PESTLE factors to make a more informed decision. This is because transport operators, city planners, and policymakers also consider other political, legal, technological, and environmental concerns for promoting a vehicle mode. Hence, to cater to all stakeholders and capture the plethora of factors considered in this paper, a multistakeholder perspective is adopted without confining to a single perspective. The research objectives or motives of this paper are defined as follows:

1.1. Research Motives.

To identify and evaluate the significant factors from the internal and external environments that influence India's transport mode decision

To evaluate and identify the best-suited transport modes from a multistakeholder perspective (benefitting passengers, operators, planners, and policymakers) using a hybrid multicriteria model

To provide practical and policy implications to the stakeholders

Since there are many factors/criteria to be considered which may involve evaluating the benefit-cost analysis of the elements, a multicriteria [3–6] model is formulated. The model evaluates the trade-off of different factors influencing the vehicle selection decisions to provide suitable weightage to prioritize the elements and identify the most feasible alternative. The order of precedence of factors may vary from passenger to passenger, and hence, the model factors in priorities of a single passenger scenario. In such circumstances, there is a need to have intelligent decision systems that can evaluate the criteria as well as choose the best alternative, among others. Thus, in the process, a Hybrid Multicriterion Fuzzy AHP-TOPSIS/Fuzzy TOPSIS/EDA/Ensemble Ranking model [7, 8] is simulated. The model evaluates the set criteria and the available alternatives after considering the preference of passengers extracted from primary data and assigns weights to the factors that are input for the ranking of vehicle transport modes. The rest of the paper is structured as follows. The literature review is elucidated in Section 2. The research methodology adopted for the hybrid multicriteria model is discussed in Section 3, followed by the proposed model and data collection in Section 4. The application of the proposed model is then illustrated and analyzed in Section 5. Finally, the study's implications are presented in Section 6, and the investigation is concluded in Section 7. The references are then stated.

2. Review of Literature

The review of the existing literature involves identifying the factors and subfactors influencing the vehicle mode selection (Section 2.1) and reviewing the existing research in the domain (Section 2.2).

2.1. Literature Review of Factors and Subfactors Influencing Selection of a Transport Mode. The existing studies that analyze the passenger preferences categorize the factors into (i) Political factors, (ii) Economic Factors, (iii) Social Factors, (iv) Technological Factors, (v) Legal Factors, and (vi) Environmental factors [2]. Political factors are further subdivided into Political Stability and Government Policy. Economic factors include Duties and Taxes, Economic Growth, Unemployment, and Cost Efficiency, while Social Factors include Health, Safety, and Security. Technological factors define vehicle metrics like Maintenance and Fuel Efficiency, while the Legal factors are Restricted Movement and Legislation. Environmental factors are Air Pollution and Noise Pollution, as depicted in Figure 1.

The factors and subfactors that influence the selection of an available transport mode are as follows:

2.1.1. Political Factors [9]. The political situation is a critical factor that determines the choice of transport. The enforcement and amendments made to significant transportation policies, including employment laws for staff, environmental legislation, vehicle taxation laws, health, and safety norms [10], decide as to which transport mode is feasible. The political factors are further subdivided into

(1) Political stability [10]: political stability is a measure of the state of political affairs in a region and the ruling political party's power. Political disputes may impact routes' functioning and may disrupt the traffic on the roads; hence, the mode of transport is affected. This is common in metropolitan and politically active cities in India, impacting vehicles' operation due to road strikes.

(2) Government policy: the government policy issues refer to the introduction of competing for public

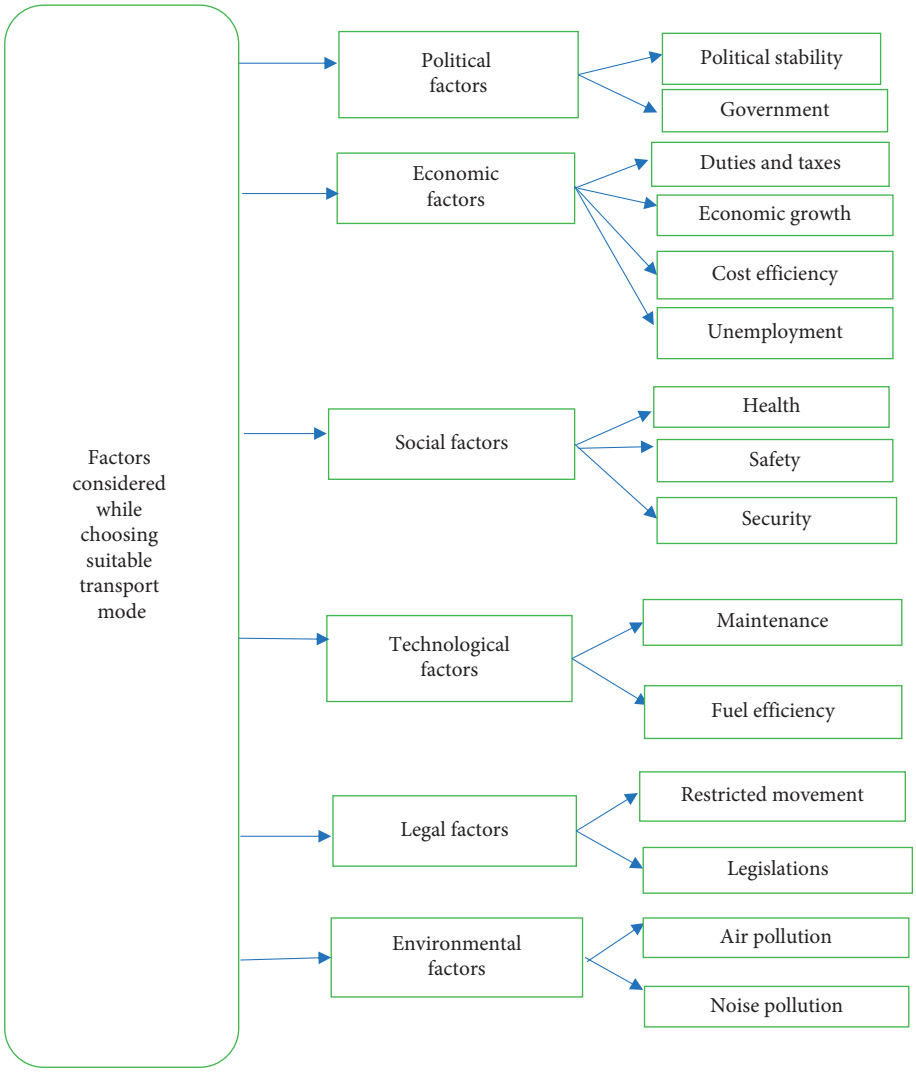


Figure 1: Factors that influence the vehicle mode selection.

transports and the implementation of newer unconventional modes of transport. The Ministry of Transportation and Roadways promulgates legislations to different route modes of transport, which impacts the city's route landscape and, in turn, the choice of transport.

2.1.2. Economic Factors [11]. However, another set of factors that influences the feasibility of transport modes are economic factors. The financial issues include

(1) Duties and taxes [12]: the duties and fees levied on vehicles like toll tax, road tax, etc., impact the cars' functioning across the allocated routes. Passengers prefer routes and modes of transport with minimal charges levied on the vehicles, which influences their decision-making.

(2) Economic growth: the growth of the economy influences lifestyle changes, and this, in turn, impacts the choice of the transportation mode.

(3) Cost efficiency [13]: cost efficiency is a metric that quantifies the cost incurred for adopting a mode of transport, viz-viz, the distance and time taken to travel to a place across a particular route. Routes and modes of transportation that are more cost-effective are, therefore, preferred.

(4) Workforce utilization/unemployment rate: the transportation industry is infrastructure-dependent and highly labor-intensive. Alternative transport modes especially are highly dependent on the low-cost labor market. At the same time, conventional transport modes cannot be completely phased out since the livelihood of the drivers and the staff (employees) will be disrupted.

2.1.3. Social Factors [14]. Social factors include attributes like demography, education level, and income with the perception of health, safety, and security [15], which directly impact the transportation industry. Social factors are grouped into the following subfactors in the paper:

(1) Health: the health conditions and issues faced by passengers and drivers influence the continuation of transportation modes' operation.

(2) Safety: the safety of the vehicle and the route adopted are both critical to choose a suitable vehicle mode.

(3) Security: the measure of the risk exposure of the passengers and the staff to criminal activities like physical harassment, robbery, stealing, etc., is security. Passengers intuitively prefer highly secure modes of transport.

2.1.4. Technological Factors [16]. The evolution of technology developed ways to quantify the maintenance of vehicles. There are several technological factors of which the factors "upgradation in fuel technology" and the "rate of fuel efficiency" [17] are significant drivers of choosing the best mode of transport. The operation and the maintenance of the various kinds of unconventional transport modes are technology dependent, and their performance is quantified in terms of fuel efficiency. This, in turn, prompts the need for fuel technology upgradation. Furthermore, this factor is also intuitively considered necessary at a layman level to choose a particular vehicle mode and is hence incorporated in the multicriteria model to provide customized vehicle alternative recommendations. Technological factors define vehicle metrics like Maintenance and Fuel Efficiency as follows:

(1) Maintenance: the extent to which a vehicle is damage-free and is in good condition for operability on the roads is called maintenance. Maintenance is also related to the safety of the vehicle and is of high importance.

(2) Fuel efficiency: the distance travelled by a vehicle viz-a-viz the fuel consumed to complete the trip is a measure of how fuel-efficient the vehicle is. The higher the value, the more economical the vehicle will be.

2.1.5. Legal Factors [18]. Laws governing the transport authorities and vehicle plying policies are essential to determine the transport mode. Legal factors can be:

(1) Restricted movement: the legal factors deal with regulatory operating bodies, statutory rules, and the policies regarding "Restricted Movement" [19] considered to analyze the feasibility of a vehicle mode. Banning some unconventional modes of transport as well as imposing restrictions on the authorities in their operation on specific roads.

(2) Legislation: the traffic routing policies and the extent of stringent measures taken to administer the functioning of the unconventional modes of transport also influence vehicle mode choice.

2.1.6. Environmental Factors [20]. The ecological dimension is captured by the factor of environmental pollution [21]. Pollution is subclassified into:

(1) Air pollution: the vehicle modes that release poisonous pollutants into the air like carbon monoxide and nitrous oxides have a damaging effect on the environment leading to hazards like global warming, acid rain, and ozone layer depletion if not curbed at early stages.

(2) Noise pollution: the vehicles also can contribute to a major portion of the community noise, particularly disruptive in the residential areas. The traffic flow speed is directly proportional to the number of decibels of noise emitted and needs to be controlled.

The existing studies on vehicle alternative selection are discussed below:

2.2. Existing Research on Vehicle Alternative Selection. Christiansen [9] studies how political factors impact the choice of transportation mode. The Hierarchial Ordinary Least Squares technique has been adopted to test the hypothesis. It can be inferred that political considerations are significant, and thus, policies need to be analyzed in further detail to choose a suitable vehicle mode. Zhou and Zhang [12] examine the impact of environmental factors on the vehicle mode to opt for. A systematic literature review is performed. The fuel efficiency and minimization of carbon emissions are essential factors to consider.

Gössling [22] investigates the challenges of introducing e-scooters in ten major cities by content analysis of Internet searches, print media, TV, and radio websites. Policy concerns related to the implementation of this transport mode are assessed. The paper recommends urban planners to introduce policies regarding maximum speed, bicycle infrastructure, and limit the number of licensed operators.

Daisy et al. [23] advocate a cohort-based approach to analyze the patterns of trips, to propose a suitable transport mode choice factoring in socio-demographics, trip attributes, and land-use patterns using a multinomial logit model. It is found that sociodemographic characteristics and tour attributes are significant determinants of travel behavior.

Ashmore et al. [24] utilized two tenets of the Hofstede cross-cultural indices—power differential and individualism versus collectivism—to develop the transport choice model through qualitative deductive thematic analysis. The significant differences between the cross-cultural group symbolizes the importance of symbolism and culture in transport choice.

Tarabay and Abou-Zeid [25] investigate the future of ride-sharing services like Uber and Careem by the American University of Beirut, Lebanon. They developed a hybrid choice model that predicts the choice between traditional modes of transport to ride-sourcing services for social/recreational trips in Lebanon. It is found that the factors influencing the switching choice are: door-to-door travel time, pickup waiting time, one-way fares, and individual differences in perceptions and attitudes toward ride-sourcing services.

Van Ristell et al. [26] analyze the traffic and environmental factors impacting school-going children's transport choice in England. Multinomial logit modelling and mixed multinomial logit modelling are adopted. It is suggested to advocate the "nearest school" policy, wherein students are admitted to schools in their proximity to prevent the rise in carbon-dioxide emissions and for cost-effectiveness.

Stoilova [16] studies the influence of technological factors on why some vehicle modes are preferred. Multicriteria methods like Fuzzy AHP and Promethee are adopted. Factors like fuel efficiency and updating fuel technology are found to be significant. Tian et al. [21] evaluated the impact of pollution on choosing a vehicle mode. A hybrid SBM-DEA model is adopted. Vehicles with the least impact on pollution are preferred.

Roorda et al., [27] examine the factors influencing the choice of the transport mode. The nested Logit approach is implemented. There is a need to adopt safer and eco-friendly vehicle modes.

Jian et al. [28] aim to identify the factors that influence vehicle mode selection. The Spatial Hazard based model is adopted. The model with the best fit is chosen.

Luo et al. [29] study the importance of environmental factors. Model Predictive Control (MPC) framework is adopted. It is found that the most fuel-efficient vehicle is to be chosen, emphasizing the importance of environmental factors.

Chen and Wang [30] study how political factors like taxation laws impact transportation mode choice. Stochastic processes are adopted. Policies need to be analyzed differently for variable demand in further detail to choose a suitable vehicle mode.

Onstein et al. [18] aim to determine the role of legislations governing vehicles. This paper proposes a Best-Worst method. Legislations play an essential role.

To study how fuel efficiency impacts vehicle mode, binary logit models were adopted by Krishna et al. [17]. Their study revealed that the more the vehicle is fuel-efficient, the higher is the probability of being preferred.

Böcker et al. [31] investigate the transport mode choices of older people in the Netherlands. Zero-inflated negative binomial models and multinomial logit regression models are adopted. Older people are encouraged to use environmentally friendly transport modes.

Furthermore, the viability of an unconventional mode of transport, namely, e-bike share vehicle, is assessed by Campbell et al. [32] in China. A multinomial logit model is used. E-bike-share is found to be a more viable mode of transportation than bikes, an already existing transport mode in China.

Chee and Fernandez [33] study the transport mode choice in Penang, Malaysia. Surveys and questionnaires are used for the qualitative study. Private transport is found to be the predominant choice for commuters.

Donald et al. [34] analyze the psychological factors influencing the transport choice. Structured Equation Modelling was adopted. The use of private transport modes like the car is driven by intention and habit, while public modes are influenced solely by purpose.

Gebeyehu and Takano [35] examine that the transport mode choice in Addis Ababa is being reviewed. The ordered logit model was adopted. Fare, convenience, and frequency were the main factors for the bus as the transport mode in Addis Ababa.

The transport preferences of school-going children were analyzed by Kamargianni et al. [36]. A multinomial probit approach was adopted. The value for more eco-friendly alternatives like bicycling is emphasized over the bus.

Madhuwanthi et al. [37] analyze the travel behavior of Srilankan citizens. Multinomial Factor Analysis is implemented. Income, vehicle ownership, safety, and comfort are the crucial factors determining vehicle choice.

Transportation mode choice is investigated in Kharagpur and Asansol by Majumdar et al. [38] Exploratory Factor Analysis (EFA) is adopted. Bicycle is found to be a feasible alternative mode.

Santos et al. [39] examine the factors that influence the modal split in European countries using regression analysis. A significant negative correlation is found between private vehicle modes like cars and public transport (bus). The transport mode patterns for young adults are investigated [40]. Content analysis was performed using N-Vivo. The use of private transport modes like the bicycle is encouraged.

The above studies adopt existing choice models to identify the best-suited vehicle mode in respective country scenarios. However, there are some limitations in the choice of the methodology used.

These limitations are thus discussed below:

2.3. Research Gap. The above studies consider single or multiple factors for determining the choice of vehicle selection. However, a multifactor approach is not adopted to analyze the cumulative impact of all the factors considered for the transport choice decision. Determining the essential factors will help take proactive steps by road transport authorities to provide transport services at subsidized rates to retain frequent passengers.

Figure 2 illustrates the limitations in the existing studies. Firstly, the vehicle mode selection problem is not explored in an Indian metropolitan city context where there are multiple conventional and unconventional transport modes, and there is a need to understand the commuter's decision to choose the relevant mode. Furthermore, there is an interplay of multiple factors by conflicting stakeholders that was not captured in such a context.

Secondly, since a single factor is not sufficient to decide on the most feasible transport choice, there is a need to implement a multifactor approach to capture commuters' personalized preferences. Furthermore, existing choice models only capture the economic perspective of the choice of transport mode while other dimensions like political, social, and ecological factors are not considered. Furthermore, in a multistakeholder and hierarchical scenario where multiple conflicting objectives are applicable, there is a need for a more robust model to capture all the factors influencing the choice of the vehicle mode. Though there are some existing multicriteria approaches to make such decisions,

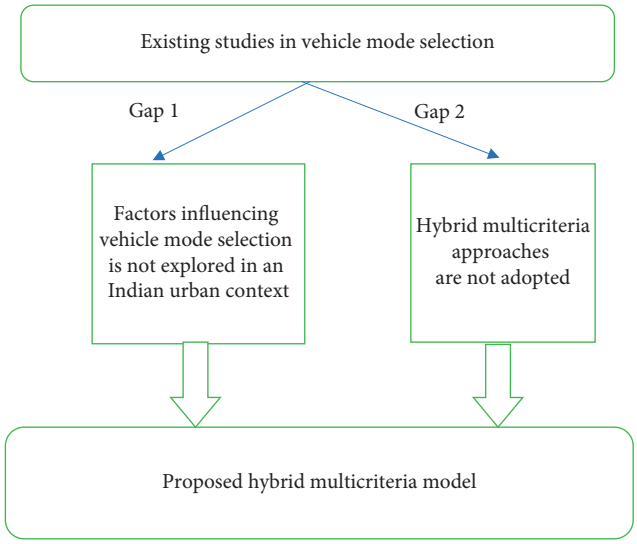


Figure 2: Research gaps.

there is a need to validate these approaches' outcome and perform a sensitivity analysis for further whetting the results. A hybrid multicriteria model that computes weights shows ranking, and validation using different multicriteria methods can provide a more optimized decision.

For overcoming the above limitations, a hybrid multicriteria model is adopted in the paper. The data collection procedure and research methodology adopted are discussed in Section 3.

3. Data Collection and Methodology

This section elucidates the data collection procedure (Section 3.2) and the research methodology adopted (Sections 3.3–3.6) in the paper. The rationale for choosing the method is detailed below:

3.1. Rationale for Methodology (PESTLE Framework and Hybrid Multicriteria Model). Existing frameworks like the SWOT capture the strengths, weaknesses, opportunities, and threats but do not factor in the ecological and the social dimensions. The different transport's survivability is driven by several external factors like government policies, carbon footprint, unfavorable laws, and constantly updating technologies. The PESTLE framework [41] is highly suitable and is adopted to capture this plethora of dimensions.

PESTLE Analysis framework stands for, Political, Economic, Social, Technological, Legal, and Environmental factors within the system to aid decision-making.

Furthermore, although there are existing techniques like choice models [36] to identify the best alternative, in the above scenario of choosing a suitable vehicle alternative mode, hybrid multicriteria methods are adopted. This is because choice models only provide alternatives from the point of view of minimizing costs. However, other dimensions like social, environmental, and political dimensions are not captured to provide the best alternative by choice models. This is needed in the above scenario for effective implementation of the PESTLE framework.

Secondly, there is a need to identify a suitable vehicle alternative from a host of conflicting objectives across stakeholders and across multiple hierarchy levels (criteria are further subdivided into subcriteria).

Thus, the hybrid multicriteria model is recommended over state-of-the-art choice models [26] and hence adopted in the paper.

Fuzzy AHP is adopted to compare the factors and alternatives by computing fuzzy scores assigning weights to the criteria. This methodology is very useful for multiple criteria decision-making in uncertain environments where the relative importance of factors influencing the transport mode cannot be assigned a crisp score. However, a fuzzy score with lower, medium, and higher bounds can be estimated for the factors. Hence, Fuzzy AHP is adopted. This Ranking is accomplished by Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [42] by structuring the problem, conducting analysis, and enabling comparison and ranking. TOPSIS decides that the best-ranked solution is one that is geometrically closest to the positive ideal solution (PIS) and farthest from the perfect negative solution (NIS). However, in scenarios where accurate comparisons cannot be handled, the fuzzy TOPSIS model is adopted to rank the vehicles. To perform maximization of beneficial factors and minimize the non-beneficial factors, the Evaluation Based on Distance from the Average (EDA) approach for ranking of vehicles is applied.

Overall, to present a consolidated picture of the various rankings and to arrive at a robust consensus ranking to choose the most feasible vehicle mode, the novel Ensemble Ranking [43] technique is adopted.

Thus, a hybrid approach [44] is envisaged to accomplish the objectives of this paper.

The methodology adopted is divided into the following phases:

3.2. Phase 1: Data Collection. To analyze the alternative modes of transport in the metropolitan city of Hyderabad, initially, the data collection phase was initiated. A "complete participation" interview approach [45] was adopted where live interviews with the passengers and drivers were conducted in the areas defined above to develop an understanding of the traffic flow statistics, day-to-day operations, and various merits and demerits of the modes of transports. To corroborate the findings, one-to-one interviews were conducted with the various stakeholders–the transport authority staff, drivers, and frequent passenger and logistics and operations researchers to identify the significant issues. The drivers and other stakeholders across the particular routes are selected. Inputs are from legal authorities from the head office of the Road Traffic Authority (RTA) situated in Hyderabad, Telangana. The expert academics whose participation was solicited were top authorities on operations research. The interviews posed categorical questions initially to the stakeholders to give their opinions on which transport mode is better based on criteria like commuting time, fuel

efficiency, and environmental friendliness on a scale of 1–9. The questionnaire is enclosed below in the Appendix.

The above responses provided by the passengers are then transformed to fuzzy scores according to the fuzzy scale provided in Table 1 for computing weights by Fuzzy AHP since this technique uses only fuzzy inputs and not crisp numbers.

The sample involves 300 passengers belonging to different localities, age groups, professions, and gender. Out of the total sample, there were 200 male and 100 female respondents. Almost 23.4 percent of the respondents were aged between 30 and 50 years, while the majority of the commuters are in the age group of <30 years and constitute 76.6% of respondents. Table 1 illustrates the sample demographics statistics.

Table 1: Sample demographics.

Gender	Number	Percentage
Males	200	66
Females	100	34
Age		
<30 years	230	76.6
30–50 years	70	23.4
Nature of locality		
High class	10	3.3
Middle class	220	73.3
Lower middle class	70	23.4
Profession		
Private employees	120	40
Government employees	100	33.3
Students	50	16.6
Miscellaneous	30	10
Total respondents	300	

3.3. Phase 2: Factor Development. Based on their responses, the weights computed are provided as inputs to the ranking models TOPSIS/Fuzzy TOPSIS/EDA/IRP below. But before that, the next phase, phase 2, involves classification of the factors into benefit and cost factors for inputting to the ranking models TOPSIS, Fuzzy TOPSIS, and EDA. For instance, political instability is considered a cost factor, implying that the higher the value, the more disruptive it is to the vehicle mode. At the same time, a positive sign (+) is provided for fuel efficiency, which indicates that the higher the value, the more beneficial is the factor. The impacts of these criteria depicted in the criteria evaluation matrix were quantified on a 1–9 scale (Saaty scale) as follows: 1 (no effect) and 9 (highest impact), while for the alternatives, the rating score provided by the passengers was in a 1–10 scale as follows: 1 (lowest) and 10 (highest). Factors like fuel and | cost efficiency are measured per kilometer, while the operating cost is calculated on a monthly basis.

The issues were then categorized into factors and subfactors based on the PESTLE (Political, Economic, Social, Technological, Legal, Environmental) framework. The relative importance of the elements is computed by the Fuzzy AHP (Fuzzy Analytic Hierarchy Process). The best alternative mode of transport is assessed and ranked using the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution), Fuzzy TOPSIS, EDA, and IRP methods.

3.4. Phase 3: Weight Computation by Fuzzy-Analytical Hierarchy Process. The next phase, phase 3, involves computing the weights for each factor classified above using the Fuzzy Analytical Hierarchy Process (Fuzzy AHP). Before understanding the functionality of the Fuzzy AHP, the fuzzy set theory and the Analytical Hierarchy Process (AHP) are detailed below.

3.4.1. Fuzzy Set Theory. Any event, process, or function that is continuously variable and cannot be definitively categorized into true or false is said to be Fuzzy. Fuzzy logic implements the above principle to degrees of truth rather than the usual true/false or 1/0 like the Boolean logic. In fuzzy systems, the values are indicated between 0 (untrustworthy) to 1 (right).

Fuzzy Logic principles in set theory constitute Fuzzy Set Theory [46] that allows partial membership, i.e., contains the elements with variable membership in the set.

3.4.2. Analytical Hierarchy Process [47]. The Analytical Hierarchy Process (AHP) is a multicriterion approach adopted to assign relative weights to each criterion. AHP performs cost-benefit analysis based on absolute priorities [48, 49] and weightage ranking. The technique was devised by Wind and Saaty [50].

The AHP technique for computing the weights works as follows:

Step 1: the pairwise comparison criteria matrix A is constructed in the form of a $x * x$ matrix, where x is the number of factors considered in the study.

Each entry $a_{jk}$of the matrix A constitutes the relative weightage of the jth factor to the kth factor. If $a_{jk} > 1$, the jth criterion is more critical, and if $a_{jk} < 1$, the jth criterion is not considered as necessary as the kth criterion. If two criteria have the same level of importance, then the entry a_{jk} is 1.

The entries a_{jk} and a_{kj} satisfy the following constraint:

$$a_{kj} * a_{jk} = 1. \tag{1}$$

Obviously, $a_{jj} = 1$ for all j.

The relative importance between the two criteria is quantified from 1 to 9, as shown in Table 2:

The interpretations are reflective of the qualitative evaluations of how important a criterion is over the other.

Step 2: once the matrix A is built, it is possible to derive from A the normalized pairwise comparison matrix A_{norm} where entry of the matrix $a_{jk\,\text{norm}}$ is:

$$a_{jk\,\text{norm}} = \frac{a_{jk}}{\sum a_{lk}}, \tag{2}$$

Table 2: Fuzzy linguistic terms with defined triangular scales [50].

Saaty scale	Definition	Fuzzy triangular scale
1	Equally important	(1,1,1)
3	Weakly important	(2,3,4)
5	Fairly important	(4,5,6)
7	Strongly important	(6,7,8)
9	Absolutely important	(9,9,9)

where a_{lk} is the column-wise sum of entries for each criterion.

Step 3: finally, the factor weight vector w_j is built by averaging the entries on each row of A_{norm}, i.e.,

$$w_j = \frac{a_{jl\,\text{norm}}}{x}. \tag{3}$$

Fuzzy Analytic Hierarchy Process (F-AHP) [51] integrates the fuzzy concept to Analytic Hierarchy Process (AHP). The pairwise comparisons are performed by representing linguistic variables as fuzzy triangular numbers.

The linguistic terms are defined in terms of the following sets of fuzzy triangular scales:

Table 2 depicts the Saaty scale from 1 to 9 and the corresponding fuzzy equivalent (l, m, h); "l" represents the lower bound and "h" the higher bound with the crisp score and the middle value by the variable "m." For instance, a Saaty scale value of 2 indicates the weak importance of factor "i" while a value of 7 indicates high importance.

The weights computed by the Fuzzy AHP are validated by calculating two consistency ratios (CRm and CRg). These ratios are computed by the following procedure [52]:

Step 1: subdivide the fuzzy pair-wise comparison matrix (used for computing weights) into two matrices:

The first matrix is formed from the middle element "m" of the fuzzy comparison matrix denoted by A_m where: $A_m = [a_{ijm}]$.

The second matrix is derived as the geometric mean of the lower bound and the higher bound elements of the above fuzzy comparison matrix denoted by formula: $A_g = [a_{ijg}]$.

Step 2: the priority weight vector W is computed as the normalized n^{th} root of the elements in each of the above matrices:

where n is the number of criteria considered in the study and $W_i = n^{\text{th}}$ is the root value of the criterion "i" with respect to other criteria/sum of nth root values of all criteria

Step 3: the sum of product of all priority weight vector elements with their corresponding criteria column-wise sum computes the lambda-max for both the above matrices as follows:

where

$$\begin{aligned} \text{lambda} - \max_m &= \sum W_i * \text{row} - \text{wise sum of } a_{ijm}, \\ \text{lambda} - \max_g &= \sum W_i * \text{row} - \text{wise sum of } a_{ijg}. \end{aligned} \tag{4}$$

Step 4: compute the consistency index (CI) for each matrix from the corresponding lambda-max by the formulae:

$$\text{CI}_m = \frac{(\text{lambda} - \max_m - n - 1)}{n}, \tag{5}$$

for the number of criteria "n"

$$\text{CI}_g = \frac{(\text{lambda} - \max_g - n - 1)}{n}. \tag{6}$$

Step 5: compute the consistency ratio (CR) by dividing the Consistency index (CI) values by the Random Index (RI), where RI is specified by Gogus and Boucher [52] for each number of criteria(n), for instance, RI_m for the number of criteria = 3 is 0.489 while RI_g is 0.1796.

$$\begin{aligned} \text{CR}_m &= \frac{\text{CI}_m}{\text{RI}_m}(n), \\ \text{CR}_g &= \frac{\text{CI}_g}{\text{RI}_g}(n). \end{aligned} \tag{7}$$

Step 6: check for CR_m and CR_g values; if they are less than 0.1 (10%), they signify that the weights computed by Fuzzy AHP are valid, else the weights need to be checked.

3.5. Phase 4: Ranking the Viable Transport Alternatives. The vehicle transport alternatives are then ranked using the TOPSIS, Fuzzy TOPSIS, EDA, and IRP techniques detailed below:

3.5.1. Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [42, 53]. TOPSIS is a multicriteria decision analysis method, which was originally developed by Hwang and Yoon [42]. The best option depends on the distance with respect to the average solution (AV).

The proposed technique works as follows:

Step 1: define the evaluation matrix consisting of m alternatives and n criteria as follows:

$$X = \left[X_{ij}\right]_{n\times m} = \begin{matrix} X_{11} & X_{12} & & X_{1m} \\ X_{21} & X_{22} & \dots & X_{2m} \\ X_{n1} & X_{n2} & & X_{nm} \end{matrix}, \tag{8}$$

where X_{ij}denotes the decision value of the ith alternative on the jth criterion.

Step 2: the matrixes normalized to the matrix

$$X_{\text{norm}} = \frac{X_{ij}}{\left(\sum x_{kj}^2\right)^{1/2}}, \tag{9}$$

where i is from 1 to m and j is from 1 to n using the normalization method

Step 3: compute the weighted normalized decision matrix

$$S_{ij} = X_{\text{norm}} * w_j. \tag{10}$$

Step 4: calculate the worst and best alternatives A_w and A_b , respectively, as follows:

$$\begin{aligned} A_b &= \left[A_{bij}\right]_{n\times m}, \\ A_w &= \left[A_{wij}\right]_{n\times m}. \end{aligned} \tag{11}$$

if the jth criterion is beneficial,

$$\begin{aligned} A_b &= \text{MAX}\left(X_j\right), \\ A_w &= \text{MIN}\left(X_j\right). \end{aligned} \tag{12}$$

if the jth criterion is nonbeneficial,

$$\begin{aligned} A_b &= \text{MIN}\left(X_j\right), \\ A_w &= \text{MAX}\left(X_j\right), \end{aligned} \tag{13}$$

where A_{bij} and A_{wij} represent the positive and negative proximity, respectively, from the average for a particular factor "j"

Step 5: calculate the largest distance between the target alternatives A and A_w:

$$d_{iw} = \left(\sum\left(A_{ij} - A_{wij}\right)^2\right), \tag{14}$$

i is from 1, 2, 3, 4, . . ., m

Similarly, compute the best distance between A and A_b:

$$d_{ib} = \sum\left(A_{ij} - A_{bij}\right)^2. \tag{15}$$

Step 6: calculate the similarity s_{iw}:

$$s_{iw} = \frac{d_{iw}}{\left(d_{iw} + d_{ib}\right)}, \tag{16}$$

where s_{iw} ranges from 0 (worst) to 1 (best)

Step 7: the alternatives are ranked according to the s_{iw}.

The first ranked alternative is the best choice among the modes of transport.

3.5.2. Fuzzy TOPSIS [54]. Fuzzy TOPSIS is one of the best methods to get an ideal solution when there is uncertainty in the selection process. Fuzzy TOPSIS (F-TOPSIS) integrates the fuzzy concept to TOPSIS [55]. Fuzzy TOPSIS uses fuzzy triangular numbers in terms of the sets of fuzzy triangular scales defined in Table 2:

In the paper, the vehicle modes are ranked using Fuzzy TOPSIS technique in the R tool, which has a predefined package "FuzzyMCDM" that considers the weights and input matrix for ranking the vehicles.

3.5.3. Evaluation Based on Distance from Average Solution (EDAS) [56]. This technique ranks the alternatives in terms of being the closest to the best-case and farthest from the worst-case scenario (nadir) solution [57]. The best option is a function of the distance from the average solution represented by AS. The ideal solution in the proposed method is not required to be computed. The Positive Distance and the Negative Distance from Average, i.e., PDA and NDA, are calculated. Let us suppose that we have n alternatives and $m$factors. The proposed technique works as follows:

Step 1: select the best factors defining the alternatives.

Step 2: construct the alternative-criteria matrix X:

$$X = \left[X_{ij}\right]_{n\times m} = \begin{matrix} X_{11} & X_{12} & & X_{1m} \\ X_{21} & X_{22} & \dots & X_{2m} \\ X_{n1} & X_{n2} & & X_{nm} \end{matrix}, \tag{17}$$

where X_{ij} represents the i^{th}alternative on thej^{th}criterion.

Step 3: determine the mean solution AVshown as follows:

$$AV = \left[AV_j\right]_{1\times m}, \tag{18}$$

where

$$AV_j = \frac{\sum X_{ij}}{N}. \tag{19}$$

Step 4: the PDA and the NDA are computed differentially for cost and benefit criteria, shown as follows:

$$\begin{aligned} \text{PDA} &= \left[\text{PDA}_{ij}\right]_{n\times m}, \\ \text{NDA} &= \left[\text{NDA}_{ij}\right]_{n\times m}. \end{aligned} \tag{20}$$

, if the j^{th}criterion is beneficial,

$$\begin{aligned} \text{PDA}_{ij} &= \frac{\text{MAX}\left(0,\ X_j - AV_j\right)}{\text{AV}_j}, \\ \text{NDA}_{ij} &= \frac{\text{MAX}\left(0,\ AV_j - X_j\right)}{AV_j}. \end{aligned} \tag{21}$$

if thej^{th}criterion is nonbeneficial,

$$\begin{aligned} \text{PDA}_{ij} &= \frac{\text{MAX}\left(0, AV_j - X_j\right)}{AV_j}, \\ \text{NDA}_{ij} &= \frac{\text{MAX}\left(0,\ X_j - AV_j\right)}{AV_j}, \end{aligned} \tag{22}$$

where $\text{PDA}_{ij}$and NDA_{ij} denote the positive and negative distance, respectively, from the average solution for a particular factor j.

Step 5: compute SP_i and SN_i for all alternatives, shown as follows:

$$\mathrm{SP}_i = \sum w_i * \mathrm{PDA}_{ij},$$
$$\mathrm{SN}_i = \sum w_i * \mathrm{NDA}_{ij}, \tag{23}$$

where w_i is the weight of the ith criterion.

Step 6: compute the normalized values of SP and SN for all alternatives, shown as follows:

$$\mathrm{NSP}_i = \frac{\mathrm{SP}_i}{\max(\mathrm{SP}_i)},$$
$$\mathrm{NSN}_i = \frac{\mathrm{SN}_i}{\max(\mathrm{SN}_i)}. \tag{24}$$

Step 7: evaluate AS for all electives, shown as follows:

$$\mathrm{AS} = 0.5 * (\mathrm{NSN}_i + \mathrm{NSP}_i). \tag{25}$$

Step 8: the alternatives are ranked in the descending order of the average score, AS, and the highest average score is considered the best alternative.

3.5.4. Interpretive Ranking Procedure (IRP) [58]. The IRP is a ranking procedure used for validation and is adopted due to the perfect blending of rational selection processes with rudimentary intuitive processes.

The IRP is a novel ranking method [59] that ranks individual actors based on their performance outcomes viz-a-viz internal and external processes.

In the IRP, expert inputs that impact the interpretive logic for factor dominance are applied for paired comparison. The IRP makes it easy to distinguish the influence of interactions rather than the variables in an abstract sense.

The basic steps of the IRP are as follows:

(1) Categorize the variables into two sets—one to be ranked, second the criteria for ranking
(2) Identify the relationship between the two sets of variables
(3) Construct a cross-interaction matrix between the two sets of variables
(4) Interpret the binary relationships by converting to a cross-interpretive matrix
(5) Translate the matrix into a dominating interactions matrix representing the relative dominance of one actor over the other
(6) Rank the actors or alternatives based on the net dominant score

3.6. Phase 5: Validation of the Rankings. The final phase is the validation of the rankings for which the Ensemble Ranking procedure is adopted. The Ensemble Ranking technique is constructed from the above four methods (TOPSIS, Fuzzy TOPSIS, EDA, and IRP) using the methodology entailed in Mohammadi and Rezaei [43]. The weights computed above are also validated by domain experts in the field of transportation.

The methodology aims to compute a consolidated ranking system from different individual ranking systems to maximize the rankings' consensus and validity.

Consider "n" techniques which assign a rank of R_1, R_2, $R_3, \ldots, R_n$ to a particular alternative based on their respective methodology. Consider an assumed consolidated ranking of R^* for the alternative. The Ensemble Ranking technique aims to minimize each rank's Euclidean distance from the above consolidated ranking R^*.

For this purpose, a quadratic minimizer function is constructed as

$$\min \frac{1}{2} * \sum_{n=1}^{N} (R_n - R^*)^2, \tag{26}$$

where R_n is the individual computed rank, and R^* is the consolidated assumed ranking. The above function aims to maximize the consensus to consolidated ranking by minimizing the distance to the consolidated ranking.

An optimal weighted Ensemble ranking procedure is formulated by assigning individual weights $w_1, w_2, \ldots, w_n$ to each of the procedures and changing the weights iteratively till a convergence to the final solution is reached.

Each auxiliary variable representing each individual ranking method is denoted by α; where

$$\alpha 1 = \mu * (R1 - R^*)^2 \text{ [representing minimize function for rank method 1].} \tag{27}$$

The weights are computed as the normalized form of auxiliary variables by the formula:

$$w_n = \frac{\alpha_n}{\sum_1^N \alpha_n}. \tag{28}$$

The consolidated ranking is the sum-product of each rank with respective weightage:

$$R^* \text{ (optimal)} = \sum_n w_n R_n. \tag{29}$$

Figure 3 summarizes the methodology adopted by this study below:

The results are enclosed in Section 5. The proposed multicriteria model explaining the relationship between the criteria and the alternatives based on the nature of vehicle is illustrated below:

4. Proposed PESTLE Model

The model determines the best vehicle mode constructed based on a tree structure with level 1 (top-most root) as the

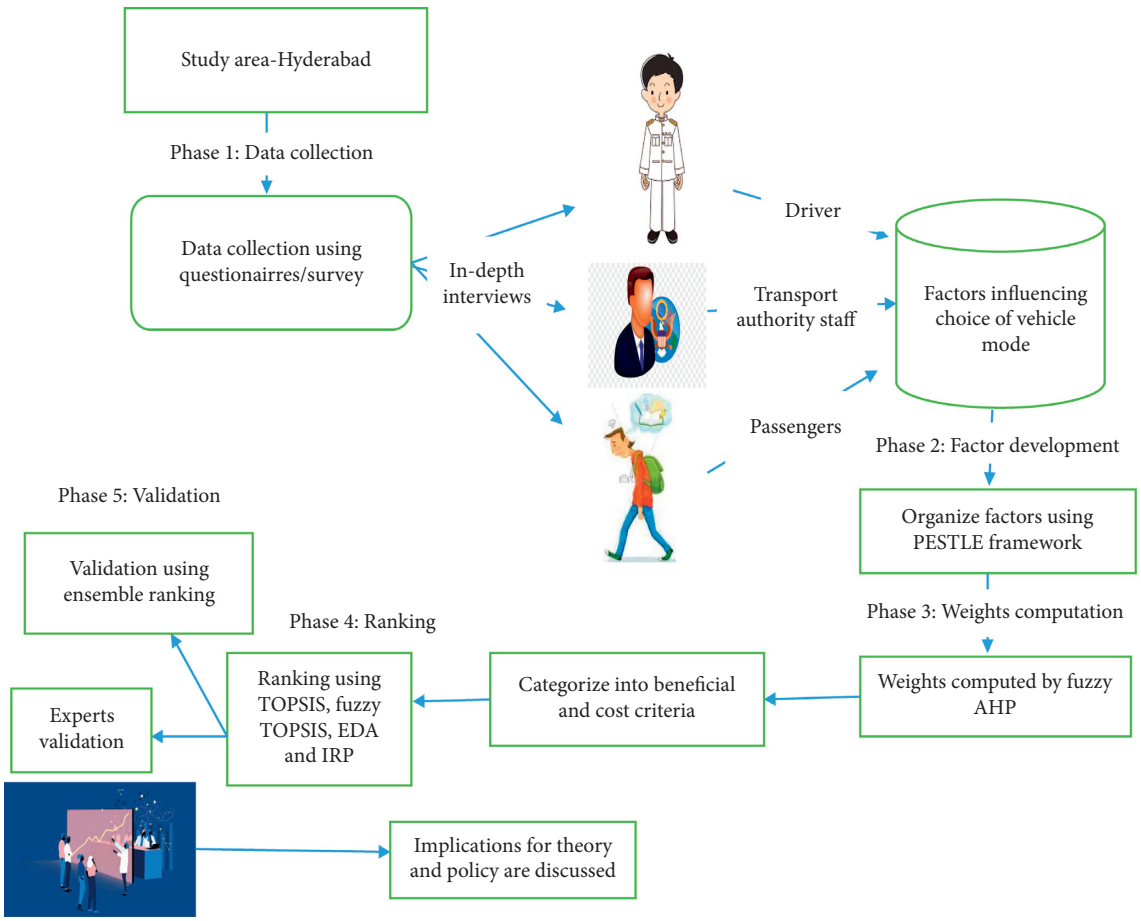


Figure 3: Methodology adopted for the study.

multicriteria objective of choosing the most feasible and viable transport mode alternative for passengers. Secondly, the vehicle alternatives are specified in level 2, and criteria in level 3.

Figure 4 illustrates the proposed hybrid model to examine the criteria for choosing a vehicle mode for the passengers among alternative options (defined in Level 1). The criteria considered are: Political, Economic, Social, Technological, Legal, and Environmental factors (subfactors are illustrated in Figure 1). These are the criteria that the passenger considers for choosing the best vehicle alternative. The Fuzzy AHP is initially used (placed at Level 3) to separately perform pair-wise comparisons for factors and the alternatives, and to compute the fuzzy weights or priorities assigned to the criteria. Furthermore, the criteria are grouped into beneficial and nonbeneficial criteria, and these are applied on four vehicle modes, namely, Metro, Car, Public bus, and bike. The four vehicle alternatives are then ranked according to three methods: TOPSIS, Fuzzy TOPSIS, EDA, and the IRP techniques and compared (shown in Level 2). They are then validated by the Ensemble Ranking technique. The next section discusses the application of the proposed model in the context of this paper.

5. Application of the Proposed Model and Results

5.1. Study Area and Scope. Hyderabad, the capital of Telangana, the newly formed 29th state of India, has been selected as a case in point for metropolitan cities. The population is 13 million, of which around 13% are below the poverty line and earn an income of Rs. 4 Lakh per capita [60]. The population density is reported to be 18,480 per sq.km. The city generates around 45,000 motorized vehicle trips per day [61].

The suitability of the vehicle modes is not evaluated in terms of absolute importance. They are evaluated from the point of view of relative applicability in Hyderabad.

This study was conducted on the busy route connecting Nagole, an eastern residential suburb in the city outskirts of Hyderabad, to the other end of the city, Hitech City, the financial and technological hub employing millions of software professionals and analysts. Figure 5 illustrates the map of Hyderabad and a picture illustrating the routes.

This study considers four modes of transportation, locally known as–Metro train, Public bus, Bike, and Car as alternatives. Bike is a two-wheeler micro-mobility vehicle mode controlled by a handlebar. Rickshaws constitute 30% of the terrestrial vehicles and 23% of commuters with an

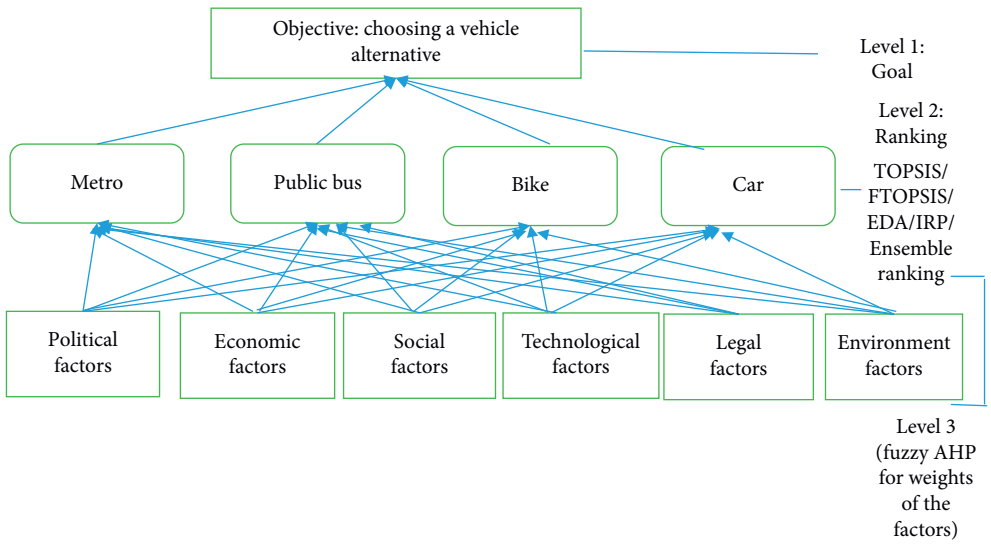


Figure 4: The proposed model.

Figure 5: The modes of transport, their routes, and the study area (in clockwise order, public bus, metro train, bike, and car).

average trip length of 8.76 km [62]. A bus is a four-wheeled vehicle carrying passengers by road on a fixed route and driven by a bus driver. Bus comprises almost 70% of the road transport [63] and is currently an accessible and frequent transportation mode for commuting within the city and is routed throughout the city radius. A car comprises about 20–30% of the road transport and is a less economical means of transportation. A metro train is a transit train that is customized for metropolitan cities. It is currently new to the city of Hyderabad. It has a daily occupancy of 4.62 lakh passengers [64] and covers the 30-km stretch from Hitech City to Nagole (the route considered for the study). A snapshot of the route and the vehicles discussed above is illustrated in Figure 5:

The phases illustrated above in Methodology are implemented below for the scenario:

5.2. Computing Weights Using Fuzzy AHP [65]. The criteria are defined and the pair wise matrix is evaluated with relative weightages assigned to each ordered tuple of criteria (C_i, C_j), where $i < j$ for all 6 factors defined above in the PESTLE framework: Political, Economic, Social, Technological, Legislative and Environmental factors subdivided into the 13 subcriteria: Political Stability, Government Policy, Duties and Taxes, Economic Growth, Unemployment, Cost Efficiency, Health, Safety, Security, Maintenance, Fuel Efficiency, Restricted Movement, and Pollution. This helps in evaluating the relative criteria significance.

The weights of all criteria computed by Fuzzy AHP is illustrated in Table 3, where the Fuzzy set weights Si, the crisp weightages, and the relative normalized weights of the factors are depicted.

Table 3 illustrates the normalized weights for each of the factors computed from the Fuzzy AHP. It is found that Political factors are the most important, followed by Social factors and Technological factors and then Economic Factors. This is because for the practical implementation of a new transport mode (in this case, metro), the political clearances are initially sought, followed by examining the impact of the new transport mode on society. Subsequently, the technological know-how is sought to construct the infrastructure while taking care of the economic and environmental factors.

The weights computed above are validated for consistency by computing the consistency ratios as elucidated above in methodology Section 3.4.2, and the values of lambda-$\max_m$ and lambda-$\max_g$ are first computed as: lambda-$\max_m = 6.22$ and lambda $- \max_g = 6.11$.

Furthermore, the consistency index values are: $CI_m = 0.044$ and $CI_g = 0.022$.

Furthermore, for the number of criteria = 6 (in study), the corresponding Random index (RI) values according to Gogus and Boucher [52] were specified to be $RI_m = 1.19$ and $RI_g = 0.38$.

Therefore, the consistency ratio values CR_m and $CR_g = CR_m = CI_m/RI_m = 0.044/1.19 = 0.0368$ (3.7%) $CR_g = CI_g/RI_g = 0.022/0.38 = 0.0578$ (5.8%)

Both ratios are less than 0.1 (10%), which implies that the weights computed by the above Fuzzy AHP procedure are valid.

Furthermore, to analyze the subcriteria weights, the pairwise comparison matrices for all the subfactors were similarly constructed to calculate their relative contribution toward the main factors.

Table 4 presents the results of the relative weights of all the subfactors.

It is found that Political Instability (Political Factor) is the major barrier followed by Safety (Social Factor), Government policies (Political Factor), and Fuel efficiency (Technological Factor). This corroborates the above findings in Table 4, indicating that Political Instability, which quantifies the authority of the political party, is most essential for the successful implementation of a new transport mode followed by social factors like safety, government policies, and technological factors like fuel efficiency.

5.3. Ranking of the Vehicle Modes. First, the weights are provided as input to the TOPSIS model after categorization into beneficial and nonbeneficial subcriteria. For instance, the subcriteria Political Instability, Government Policies, and Pollution are nonbeneficial and are considered as negative drivers for choosing the vehicle alternative while Fuel Efficiency, Cost Efficiency, and Employment are positive drivers or beneficial factors. Subsequently, the Positive and Negative Distance from the ideal solutions, i.e., the PIS and the NIS, were calculated through the weighted normalized decision matrix. As these ideal solutions represent the hypothetical scenario, the distance of each alternative from these extremes was calculated (Di+ and Di-). Based on these distances, the relative closeness index, Ci, was computed to rank these alternatives using TOPSIS in Table 5 finally.

Similarly, the weighted criteria and alternative evaluation matrix are input in *R* for computing the rank using Fuzzy TOPSIS, and the results are illustrated in Tables 6 and 7:

The evaluation matrix above represents the weights assigned to the criteria and the alternative scores (out of 10) assigned for each of the alternative transport modes; for instance, Metro is assigned a score of 10 for Technological Factors. In contrast, the Public bus is assigned a score of 7, Bike a score of 5, and Car a score of 8.

The above responses average the rating provided to the subcriteria under each of the criteria based on questions asked to the passengers (questionnaire of interview enclosed in Appendix). For instance, the above response 10 for Political Factors is the aggregated average of the rating provided to each of the Political Factors, i.e., Political instability and Government Policies (out of 10) by the passengers. The aggregated Evaluation Matrix in Table 6 for the factors is thus represented for the sake of brevity.

The same evaluation matrix in Table 6 is used as the input to the EDA model, and the PDA and the NDA were computed. Then, the weighted sum product of the PDA and the NDA, namely, SPi and SNi , respectively, were calculated for all the alternatives. After normalizing the weighted scores to NSP_i and NSN_i, the final appraisal score (AS_i) was calculated to rank the alternatives in Table 8:

The Interpretative Ranking Procedure (IRP) is implemented based on the SAP-LAP framework (Situation-Actor-Processes- Learning- Actions-Performance) where the above vehicle alternatives are considered as actors that are mapped to processes as shown in the framework in Table 9:

The SAP-LAP framework in Table 9 highlights the Situations, Actors, and Processes involved in this paper. The Situations are classified into external and internal situations, namely S1, which represents the growth of unconventional transportation modes in India metro cities, and S2, which is characterized by strong technological developments. Actors are the different external alternatives defined as A1 (Car), A2 (Bike), A3 (Metro), and A4 (Public bus). Processes are internal (P1- Technology and Business Strategy Alignment) and external (P2-Offering feasible transport alternative to commuters).

Table 3: Weights computed from the fuzzy AHP.

Criteria	Fuzzy set (Si)	Weightages	Normalized weights
Political factors	(0.198, 0.329, 0.523)	0.349	0.388
Economic factors	(0.065, 0.102, 0.162)	0.121	0.121
Social factors	(0.111, 0.192, 0.323)	0.209	0.231
Technological factors	(0.070, 0.111, 0.181)	0.110	0.134
Legal factors	(0.033, 0.049, 0.081)	0.054	0.059
Environmental factors	(0.035, 0.053, 0.091)	0.059	0.066

Table 4: Relative weights of the subfactors.

Criteria	Subcriteria	Weights of subcriteria
Political factors	Political instability	0.187
	Government policies	0.101
Economic factors	Duties and taxes	0.032
	Economic growth	0.034
	Employment	0.049
	Cost efficiency	0.082
Social factors	Health	0.052
	Safety	0.104
	Security	0.078
Technological factors	Maintenance	0.079
	Fuel efficiency	0.090
Legal factors	Restricted movement	0.035
	Legislations	0.034
Environmental factors	Air pollution	0.036
	Noise pollution	0.023

Table 5: TOPSIS results for the respondents.

Alternative mode	Di^+	Di^-	Ci	Rank
Metro	0.089	0.228	0.718	1
Public bus	0.197	0.095	0.325	2
Bike	0.226	0.101	0.309	3
Car	0.225	0.096	0.299	4

Table 6: Criterion- alternative evaluation matrix.

Weights	0.388	0.121	0.231	0.134	0.060	0.066
Alternative mode	Political factors	Economic factors	Social factors	Technological factors	Legal factors	Environmental factors
Metro	4	6	9	10	9	9
Public bus	8	6	8	7	10	10
Bike	7	9	10	5	7	7
Car	8	5	7	8	5	5

Table 7: Fuzzy TOPSIS results for the respondents.

Alternative mode	Rank
Metro	1
Public bus	2
Bike	4
Car	3

The dominance matrix in Table 10 is constructed by counting the row-wise and column-wise occurrences, indicating the number dominating (D). The number dominated (B) and the net dominance calculated by the difference between D and B determines the final ranking.

Table 8: EDA results.

Alternative mode	SP_i	NSP_i	SN_i	NSN_i	AS_i	Rank
Metro	0.08985	0.23505	0.52052	0.71514	0.61783	1
Public bus	0.17261	0.04153	1	0.12635	0.56317	2
Bike	0.16389	0.02335	0.94948	0.07105	0.51027	4
Car	0.01734	0.32868	0.10044	1	0.55022	3

The overall ranking of the vehicle alternatives under TOPSIS, Fuzzy TOPSIS, EDA, and IRP is summarized in Table 11:

Table 9: Variables of SAP-LAP in the context of choosing a suitable vehicle mode.

Components		Variables
Situation	External	S1- growth of unconventional transportation modes
	Internal	S2- strong technological developments
Actors	External	A1-car
		A2-bike
		A3-metro
		A4-public bus
Processes	Internal	P1- technology and business strategy alignment
	External	P2-offering feasible transport alternative to commuters

Table 10: Dominant matrix of actors with respect to processes.

	A1	A2	A3	A4	No. dominating (D)	D-B	Rank
A1		1			1	0	3
A2					0	−3	4
A3	1	1	1		3	2	1
A4		1			1	1	2
No. being dominated (B)	1	3	1	0	5		

Table 11: Comparison of ranking from different techniques.

Alternative	EDA	TOPSIS	Fuzzy-TOPSIS	IRP
Metro	1	1	1	1
Public bus	2	2	2	2
Bike	4	3	4	4
Car	3	4	3	3

The Metro and Public bus alternatives are consistently found to be the top 2 recommended vehicle modes under all the four techniques (EDA, TOPSIS, Fuzzy TOPSIS, and IRP), the Car is found to be the next preferred alternative under EDA and Fuzzy TOPSIS, and the TOPSIS method is found to rank the Bike as the next preferred alternative.

A sensitivity analysis of the models is performed below:

5.4. Sensitivity Analysis. The sensitivity analysis is used to investigate the results' stability over a varied range of input variable values. There are 13 subfactors involved in the current study, but the analysis of over 13 weight patterns became cumbersome. Therefore, to better understand the results, the top 5 subfactors were selected for the final analysis based on their relative importance. The factors chosen for the study are Political Instability, Cost Efficiency, Safety, Government policies, and Fuel Efficiency. The results' stability is analyzed by testing the model over five different sets of weights (indicated by P1-P5) of the top 5 subcriteria. The weights of the subcriteria computed in Table 4 were varied with 5 different combinations, and the impact on the vehicle mode ranking was empirically observed from the combinations. Table 12 demonstrates the weights considered for the subcriteria.

The sensitivity analysis charts are plotted for each of the individual multicriteria methods, namely, TOPSIS, Fuzzy TOPSIS, EDA, and IRP. The vertical bars are colored blue for the metro, orange for the public bus, gray for the bike, and yellow for the car. The bars' size is inversely proportional to ranking (longer the bar, lower it is in terms of alternative ranking).

In TOPSIS, the Metro as an alternative is found to be consistently in the top 2 alternatives, the Car is found to be consistently in the 3rd or 4th position, and the rankings of Public bus and Bike are found to be less stable.

In the Fuzzy TOPSIS, Metro is again found to be stable in the top two while the other modes of transport are varying and fluctuating.

In EDA, the Metro is the most consistently superior alternative, while the next stable choice is the Bike. The ranking of the Public bus and the Car are highly unstable; similar is the case for the IRP process.

Overall, from the sensitivity analysis depicted in Figures 6–9, the Metro is found to be the most feasible alternative, while the other options differ based on the weight assigned; for instance, if Safety and Fuel Efficiency is given high weightage, the Car is preferred. In contrast, if Cost Efficiency is given priority, the Public bus is preferred as a cheaper alternative with high connectivity.

5.5. Validation of Rankings Using Ensemble Ranking. Furthermore, the above rankings are validated by the Ensemble Ranking technique, which assigns weights to each of the ranking algorithms, namely, TOPSIS, Fuzzy TOPSIS, EDA, and IRP to compute a consolidated ranking system for the tables.

Table 12: Sets of weights used for sensitivity analysis.

Weights	Political instability	Cost efficiency	Safety	Government policies	Fuel efficiency
P1	0.187	0.110	0.104	0.101	0.080
P2	0.172	0.156	0.103	0.091	0.095
P3	0.155	0.134	0.091	0.089	0.117
P4	0.194	0.121	0.106	0.137	0.105
P5	0.191	0.109	0.117	0.095	0.114

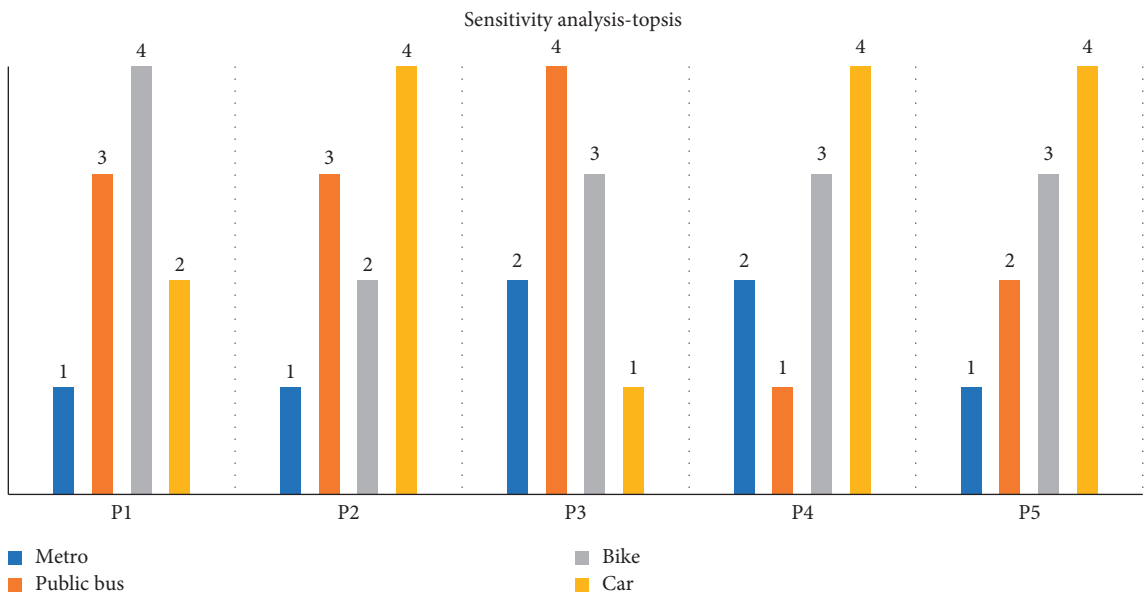


Figure 6: Sensitivity analysis using TOPSIS.

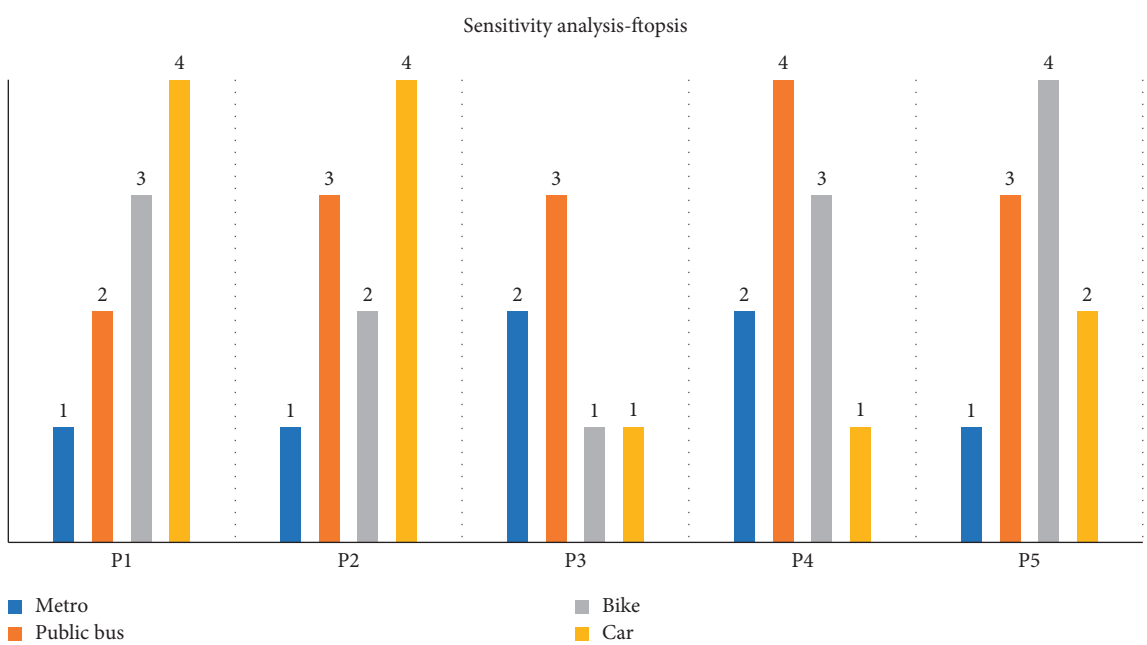


Figure 7: Sensitivity analysis using Fuzzy TOPSIS.

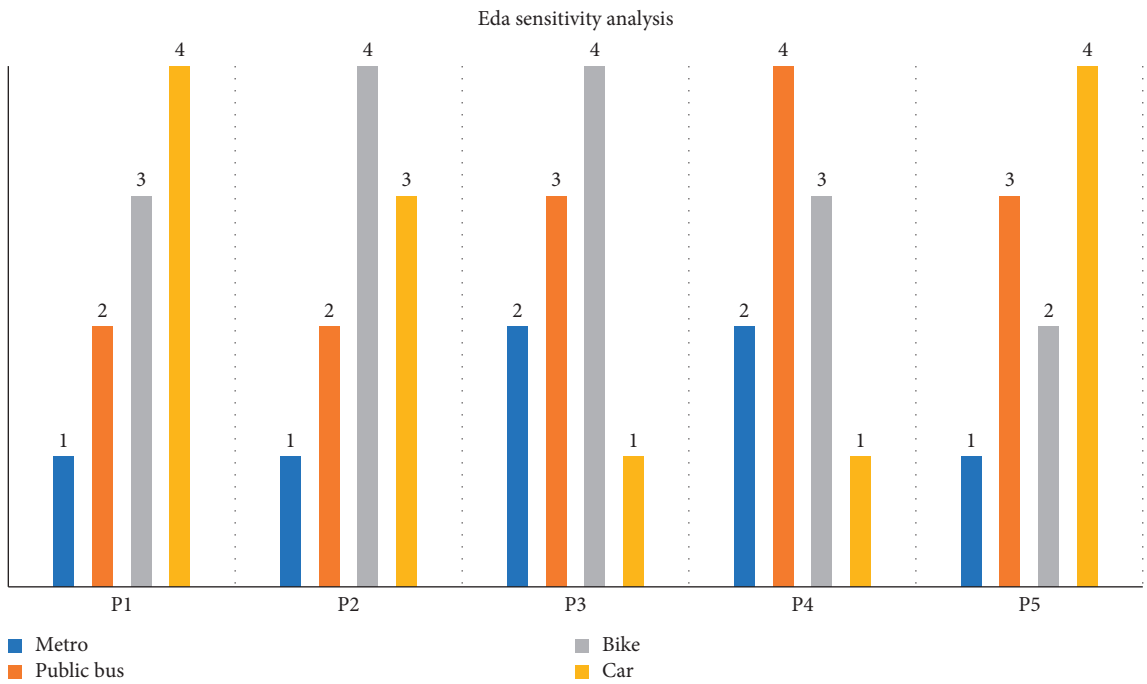


Figure 8: Sensitivity analysis using EDA.

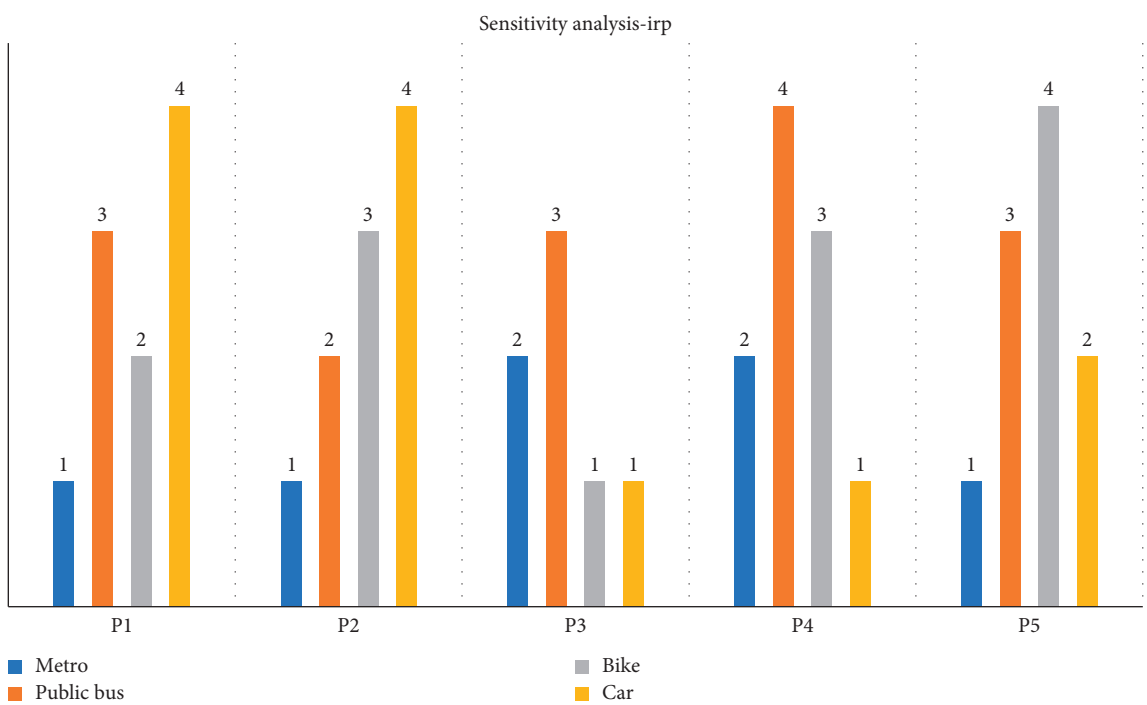


Figure 9: Sensitivity analysis using IRP.

The weights for each of the ranking algorithms are optimized using an inbuilt solver function, and the results of the Ensemble Ranking are illustrated in Table 13:

The weightages for each ranking system are optimized by an inbuilt Excel macro solver, which on clicking a button, automatically (through its inbuilt macro solver code) assigns optimal weights to each ranking algorithm ensuring that the confidence index and trust level of the algorithms are maximized [43]. EDA and Fuzzy TOPSIS are both assigned the highest weightage by the solver (33.33%), and the IRP is assigned a weightage of 27.1%.

The confidence index is a measure of extent to which all the four models, namely, TOPSIS, Fuzzy TOPSIS, EDA and IRP, are in concordance with the aggregate ranking R^*. In

Table 13: Ensemble ranking technique results.

Alternative	EDA	TOPSIS	Fuzzy-TOPSIS	IRP	R^*	Final rank
Metro	1	1	1	1	1	1
Public bus	2	2	2	2	2	2
Bike	4	3	4	4	4	4
Car	3	4	3	3	3	3
Weights computed	0.33	0.069	0.33	0.271	Confidence index Trust level	0.85 1.000

this case, the Confidence Index is 0.85, which implies that all the 4 ranking procedures are 85% in concordance with the final ranking.

Trust level metric is an indicator of the reliability of the final ranking, which is very high, i.e., 100% (1).

Overall, from the Ensemble Ranking, it can be estimated with a high confidence index and high trust level that the most feasible vehicle alternative is the Metro.

5.6. Validation of the Rankings with Experts. Having computed each of the rankings, performing a sensitivity analysis to ensure their stability, and aggregating the results with Ensemble Ranking method, the 4 vehicle modes have been ranked in the descending order based on customized preferences of the passengers filling the survey data. There is now a need to present these findings to the transportation researchers for a final whetting. A report of all the above results was sent to the research and development teams in the Road Transport Corporation to corroborate their domain knowledge about the vehicle mode applicability and the final rankings arrived at by using the multicriteria techniques.

The rankings above were corroborated with the transportation researchers in the Telangana Road Transport Corporation, and the results in Table 14 are:

It was found that the Ensemble Ranking results were consistent with the expert rankings.

This paper thus computed each factor's individual priorities influencing supply chain resilience from the relative importance values through Fuzzy AHP. From the criteria weightages and the scores assigned to each company on these criteria, the companies were ranked by TOPSIS, Fuzzy TOPSIS, EDA, and IRP. The rankings' consistency was stabilized by sensitivity analysis and aggregated to a consolidated Ensemble Ranking system with high trust level and confidence index. The final rankings were also successfully whetted by domain experts and the final ranking reveals that the Metro is the most feasible alternative. The implications are discussed below:

6. Implications of the Study

6.1. Theoretical Contributions to the Study. In this paper, a novel hybrid multicriteria model is developed to choose the most feasible transport modes wherein the Fuzzy Analytical Hierarchy Process is used to compute the weights of criteria or factors considered by passengers to select the transport mode. The weights are then utilized to rank the transport modes for which three multicriteria ranking models, namely, TOPSIS, Fuzzy TOPSIS, and EDA are adopted. The rankings are further granularly analyzed using sensitivity analysis, which examines the stability of the ranking and the sensitivity impact of criteria on the ranking system. Furthermore, the results are validated by IRP and experts. The analysis reveals that the Metro train transport mode is consistently preferred in the top 2 alternatives, while other alternatives are sensitive to the variation in weights of the criteria adopted.

Therefore, this study employs a hybrid and robust multicriteria model, which can be recalibrated and adapted in different contexts to determine the most feasible transport mode.

6.2. Implications for Practice. The implications for practice are twofold: first on the transport policy and management, and second on the passengers.

6.2.1. Managerial and Policy Implications. From the weightage computation results using the Fuzzy AHP, it is found that Political factors are the most important, followed by Social factors and Technological factors and then Economic Factors. This implies that, for the transport authority to implement a new transport mode (in this case, metro), the political clearances need to be obtained, and the political party in power should ratify the launch of a new transport mode. Second, the authority should analyze the impact of the latest transport mode on society, particularly the commuters. Subsequently, the technological know-how is to be examined by consulting operations research experts and engineers to design a transport mode with state-of-the-art technology. The economic feasibility is to be analyzed, keeping in mind the price-sensitive market of India and the commuters' economic conditions. Environmental considerations and sustainability need to be taken care of, keeping in mind the commuter's perspective.

The subcriteria weight analysis reveals that the transport authority needs to overcome the major barrier of Political Instability (Political Factor). Consequently, assuring the Safety (Social Factor) of the commuters is of high importance. The authorities need to comply with the Government policies (Political Factor) and take care of technological factors like Fuel efficiency (Technological Factor) by designing state-of-the-art transport systems in consultation with technology experts. Overall, the transport authorities need to analyze factors like safety, government policies, and technological factors like fuel efficiency for ensuring the successful implementation of the new transport mode.

Table 14: Comparison of final experts' ranking with the above-computed rankings.

Alternative	Ranking				Validation	
	TOPSIS	EDA	Fuzzy TOPSIS	IRP	Ensemble ranking	Domain expert validation
Metro	1	1	1	1	1	1
Public bus	2	2	2	2	2	2
Bike	3	4	4	4	4	4
Car	4	3	3	3	3	3

Considering the consistent performance of the metro train alternative with respect to the ranking models (TOPSIS, Fuzzy TOPSIS, EDA, and IRP), sensitivity analysis results, and validation results from Ensemble Ranking and transportation experts, it is recommended to advocate and spread awareness about the need to use the Metro as a cost-effective, environmental-friendly, safe, and fast mode of transport among all the employees, students, and other citizens.

The policy implications are thus outlined below:

Firstly, regularization and expansion of Metro routes are recommended, especially across busy and high traffic-density routes where other conventional modes of transport like Bus, Car, and Bike cause traffic jams.

Secondly, there is a need to define the hierarchy and areas of operation for the integrated use of an unconventional transport mode like the Metro and other defined conventional transport modes since eliminating the conventional transport modes with immediate action is not feasible. Busier and long-distance routes can be well-connected by the Metro. In contrast, for shorter distances and across traffic-sparse routes, conventional modes can continue to be operated to secure the livelihood of the staff of the Public bus corporations and for Bike and Bus drivers.

Thirdly, route planning can be optimized, and the use of pooling operations, especially for Cars, can be promoted to minimize the environmental damage caused by the use of private modes of transport like Cars.

Inter-modal transfer hubs can be initiated where for each metro station, a Car-pooling system can be arranged for short distances by using Car rental solutions. For instance, if a passenger (an employee of a private IT organization) needs to commute from Uppal (an eastern suburb locality in Hyderabad) to Hitech City (IT hub of the city), a Metro line from Nagole (near Uppal) to Miyapur (around 5 kilometers from Hitech City) can be routed. At the Miyapur Metro station, a Carpool rental, or a Bike or, for high traffic dense routes, a Bus can be arranged at nominal prices to safely drop the passengers at the desired location, i.e., Hitech city. This may lead to a win-win situation in the form of a public-private partnership for the Metro staff, Bus drivers, Car rental drivers, and Bike rental drivers; thus, they can collaboratively provide transport solutions to the people of Hyderabad.

6.2.2. Implications for the Society and Passengers. Passengers, primarily working professionals and students, are motivated by the results to prefer the Metro to save commuting time, for safety, and for achieving cost and fuel efficiency. Metro is particularly useful for long-distance commuting from one end of the city to the other, and proves to be politically noncontroversial, economically viable, socially safe, healthy, technologically fuel-efficient, legal, and ecologically sustainable.

7. Conclusion

This paper devises a hybrid Fuzzy AHP-TOPSIS/Fuzzy TOPSIS/EDA/IRP/Ensemble Ranking model to evaluate which vehicle alternative to choose from, keeping in mind the passenger preferences. The study focuses on the Indian metro city context with Hyderabad as a case in point.

It is hoped that this paper would benefit the passengers, transport authorities, and researchers for constructing an intelligent transportation selection model based on factors prioritized by the stakeholders using a novel hybrid multicriteria model. The simulation model provides a platform to weigh different factors developed from the PESTLE framework and select the most feasible mode of transportation. This paper is confined to a particular metropolitan city in the southern part of India, and the alternative ranking is prescribed keeping in view the local requirements of passengers and taking into consideration the relative traffic densities of the particular city of Hyderabad. However, the factors considered to evaluate the most suitable vehicle mode are applicable for all city and country scenarios, and the model can be thus recalibrated and extended to all study areas(cities) in the world with different traffic and different population densities.

The hybrid multicriteria model is developed in the context of Indian metropolitan cities, and no prior work of the subject matter dealt in this paper is found in existing studies. The subcriteria considered were the vehicle selection problem: Political Stability, Government Policy, Duties and Taxes, Economic Growth, Unemployment, Cost Efficiency, Health, Safety, Security, Maintenance, Fuel Efficiency, Restricted Movement, and Pollution. These factors were taken into account to rank the vehicle modes in the descending order. The relative importance of the criteria and alternatives is further analyzed using sensitivity analysis and is validated by Ensemble Ranking and expert decision-makers. Thus, this paper demonstrates a methodology to determine an appropriate transportation mode, keeping in mind the passengers' personalized preferences.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] "Mordor intelligence report transportation industry in India - —analysis of growth, trends and forecast (2020 - –2025)," 2020, https://www.mordorintelligence.com/industry-reports/analysis-of-transportation-industry-in-india.

[2] S. Nasrin, "Private uUniversity sStudents' mMode cChoice bBehaviour for tTravel to uUniversity: aAnalysis in the cContext of dDhaka cCity," in *Transportation Research*, pp. 299–310, Springer, Singapore, 2020.

[3] P. Vincke, "Recent progresses in multicriteria decision-aid," *Rivista di Matematica per le scienze Economiche e Sociali*, vol. 17, no. 2, pp. 21–32, 1994.

[4] K. Govindan, S. Rajendran, J. Sarkis, and P. Murugesan, "Multi criteria decision making approaches for green supplier evaluation and selection: a literature review," *Journal of Cleaner Production*, vol. 98, pp. 66–83, 2015.

[5] M. Rabbani, M. Davoudkhani, and H. Farrokhi-Asl, "A new multi-objective green location routing problem with heterogonous fleet of vehicles and fuel constraint," *International Journal of Strategic Decision Sciences (IJSDS)*, vol. 8, no. 3, pp. 99–119, 2017.

[6] M. G. Sobhani, M. N. Imtiyaz, M. S. Azam, and M. Hossain, "A framework for analyzing the competitiveness of unconventional modes of transportation in developing cities," *Transportation Research Part A: Policy and Practice*, vol. 137, pp. 504–518, 2020.

[7] G. Khatwani and P. R. Srivastava, "Employing group decision support system for the selection of internet information search channels for consumers," *International Journal of Strategic Decision Sciences (IJSDS)*, vol. 6, no. 4, pp. 72–93, 2015.

[8] M. Tajadod, M. Abedini, A. Rategari, and M. Mobin, "A comparison of multi-criteria decision making approaches for maintenance strategy selection (a case study)," *International Journal of Strategic Decision Sciences (IJSDS)*, vol. 7, no. 3, pp. 51–69, 2016.

[9] P. Christiansen, "The effects of transportation priority congruence for political legitimacy," *Transportation Research Part A: Policy and Practice*, vol. 132, pp. 61–76, 2020.

[10] K. Chen, X. Xin, X. Niu, and Q. Zeng, "Coastal transportation system joint taxation-subsidy emission reduction policy optimization problem," *Journal of Cleaner Production*, vol. 247, p. 119096, 2020.

[11] K. Kavta and B. Adhvaryu, "Walking and bBicycling to sSchool—uUnderstanding the iImpact of sSocio-economic fFactors and bBuilt eEnvironment," in *Transportation Research*, pp. 275–285, Springer, Singapore, 2020.

[12] T. Zhou and J. Zhang, "Behavioral research on transport and energy in the context of aviation," in *Transport and Energy Research*, pp. 279–294, Elsevier, Amsterdam, Netherlands, 2020.

[13] E. Kurtuluş and İ. B. Çetin, "Analysis of modal shift potential towards intermodal transportation in short-distance inland container transport," *Transport Policy*, vol. 89, pp. 24–37, 2020.

[14] S. Wang, J. Wang, and F. Yang, "From willingness to action: dDo push-pull-mooring factors matter for shifting to green transportation?" *Transportation Research Part D: Transport and Environment*, vol. 79, Article ID 102242, 2020.

[15] Y. Tyrinopoulos and C. Antoniou, "Review of factors affecting transportation systems adoption and satisfaction," in *Demand for Emerging Transportation Systems*, pp. 11–36, Elsevier, Amsterdam, Netherlands, 2020.

[16] S. Stoilova, "Methodology for mMulti-criteria sSelection of tTransportation tTechnology in tTransport nNetwork," in *Modelling of the Interaction of the Different Vehicles and Various Transport Modes*, pp. 1–103, Springer, Cham, Switzerland, 2020.

[17] A. S. Krishna, J. Thomas, and P. N. Salini, "Estimating mModal sShift of hHome-bBased wWork tTrips dDue to the dDevelopment of kKochi mMetro and rReduction in fFuel cConsumption and eEmissions," in *Transportation Research*, pp. 229–242, Springer, Singapore, 2020.

[18] A. T. Onstein, M. Ektesaby, J. Rezaei, L. A. Tavasszy, and D. A. van Damme, "Importance of factors driving firms' decisions on spatial distribution structures," *International Journal of Logistics Research and Applications*, vol. 23, no. 1, pp. 24–43, 2020.

[19] Z. Kou, X. Wang, S. F. A. Chiu, and H. Cai, "Quantifying greenhouse gas emissions reduction from bike share systems: a model considering real-world trips and transportation mode choice patterns," *Resources, Conservation and Recycling*, vol. 153, p. 104534, 2020.

[20] X. Liu, L. Gao, A. Ni, and N. Ye, "Understanding bBetter the iInfluential fFactors of cCommuters' mMulti-dDay tTravel bBehavior: eEvidence from SShanghai, China," *Sustainability*, vol. 12, no. 1, pp. 1376–13, 2020.

[21] N. Tian, S. Tang, A. Che, and P. Wu, "Measuring regional transport sustainability using super-efficiency SBM-DEA with weighting preference," *Journal of Cleaner Production*, vol. 242, Article ID 118474, 2020.

[22] S. Gössling, "Integrating e-scooters in urban transportation: pProblems, policies, and the prospect of system change," *Transportation Research Part D: Transport and Environment*, vol. 79, Article ID 102230, 2020.

[23] N. S. Daisy, L. Liu, and H. Millward, "Trip chaining propensity and tour mode choice of out-of-home workers: evidence from a mid-sized Canadian city," *Transportation*, vol. 47, no. 2, pp. 763–792, 2020.

[24] D. P. Ashmore, R. Thoreau, C. Kwami, N. Christie, and N. A. Tyler, "Using thematic analysis to explore symbolism in transport choice across national cultures," *Transportation*, vol. 47, no. 2, pp. 607–640, 2020.

[25] R. Tarabay and M. Abou-Zeid, "Modeling the cchoice to sswitch from ttraditional mmodes to rridesourcing sservices for ssocial/recreational ttrips in Lebanon," *Transportation*, vol. 47, no. 4, pp. 1733–1763, 2019.

[26] J. Van Ristell, M. Quddus, M. Enoch, C. Wang, and P. Hardy, "Quantifying the transport-related impacts of parental school choice in England," *Transportation*, vol. 40, no. 1, pp. 69–90, 2013.

[27] M. J. Roorda, J. A. Carrasco, and E. J. Miller, "An integrated model of vehicle transactions, activity scheduling and mode choice," *Transportation Research Part B: Methodological*, vol. 43, no. 2, pp. 217–229, 2009.

[28] S. Jian, T. H. Rashidi, K. P. Wijayaratna, and V. V. Dixit, "A spatial hazard-based analysis for modelling vehicle selection in station-based carsharing systems," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 130–142, 2016.

[29] L. Luo, Y. E. Ge, F. Zhang, and X. J. Ban, "Real-time route diversion control in a model predictive control framework with multiple objectives: tTraffic efficiency, emission

reduction and fuel economy," *Transportation Research Part D: Transport and Environment*, vol. 48, pp. 332–356, 2016.

[30] X. Chen and X. Wang, "Effects of carbon emission reduction policies on transportation mode selections with stochastic demand," *Transportation Research Part E: Logistics and Transportation Review*, vol. 90, pp. 196–205, 2016.

[31] L. Böcker, P. van Amen, and M. Helbich, "Elderly travel frequencies and transport mode choices in Greater Rotterdam, Tthe Netherlands," *Transportation*, vol. 44, no. 4, pp. 831–852, 2017.

[32] A. A. Campbell, C. R. Cherry, M. S. Ryerson, and X. Yang, "Factors influencing the choice of shared bicycles and shared electric bikes in Beijing," *Transportation Research Ppart C: Emerging Technologies*, vol. 67, pp. 399–414, 2016.

[33] W. L. Chee and J. L. Fernandez, "Factors that influence the choice of mode of transport in Penang: aA preliminary analysis," *Procedia-Social and Behavioral Sciences*, vol. 91, pp. 120–127, 2013.

[34] I. J. Donald, S. R. Cooper, and S. M. Conchie, "An extended theory of planned behaviour model of the psychological factors affecting commuters" transport mode use," *Journal of Environmental Psychology*, vol. 40, pp. 39–48, 2014.

[35] M.intesnot Gebeyehu and S.hin-EiE. Takano, "Diagnostic eEvaluation of pPublic tTransportation mMode cChoice in Addis Ababa," *Journal of Public Transportation*, vol. 10, no. 4, pp. 27–50, 2007.

[36] M. Kamargianni, S. Dubey, A. Polydoropoulou, and C. Bhat, "Investigating the subjective and objective factors influencing teenagers' school travel mode choice––An integrated choice and latent variable model," *Transportation Research Part A: Policy and Practice*, vol. 78, pp. 473–488, 2015.

[37] R. Madhuwanthi, A. Marasinghe, JR. P. C. J. RajapakseRPC, A. D. Dharmawansa, and S. Nomura, "Factors iInfluencing to tTravel bBehavior on tTransport mMode cChoice," *International Journal of Affective Engineering*, vol. 15, no. 2, pp. 63–72, 2016.

[38] B. B. Majumdar, S. Mitra, and P. Pareekh, "Methodological framework to obtain key factors influencing choice of bicycle as a mode," *Transportation Research Rrecord*, vol. 2512, no. 1, pp. 110–121, 2015.

[39] G. Santos, H. Maoh, D. Potoglou, and T. von Brunn, "Factors influencing modal split of commuting journeys in medium-size European cities," *Journal of Transport Geography*, vol. 30, pp. 127–137, 2013.

[40] D. Simons, P. Clarys, I. De Bourdeaudhuij, B. de Geus, C. Vandelanotte, and B. Deforche, "Why do young adults choose different transport modes? A focus group study," *Transportation Policy*, vol. 36, pp. 151–159, 2014.

[41] C. Diderich, "Understanding the iIndustry eEnvironment and iIts iImplications to sStrategy," in *Design Thinking for Strategy*, pp. 79–92, Springer, Cham, Switzerland, 2020.

[42] C. L. Hwang and K. Yoon, "Multiple criteria decision making," *Lecture Notes in Economics and Mathematical Systems*, vol. 186, pp. 58–191, 1981.

[43] M. Mohammadi and J. Rezaei, "Ensemble ranking: aAggregation of rankings produced by different multi-criteria decision-making methods," *Omega*, vol. 96, Article ID 102254, 2020.

[44] R. K. Mavi, N. Zarbakhshnia, and A. Khazraei, "A fuzzy DEMATEL analysis of cultural variables in traffic rules violation," *International Journal of Strategic Decision Sciences (IJSDS)*, vol. 8, no. 4, pp. 69–85, 2017.

[45] S. A. Mc Leod, "Observation methods," Available at: https://www.simplypsychology.org/observation.html, 2015.

[46] O. Taylan, D. Kaya, and A. Demirbas, "An integrated multi attribute decision model for energy efficiency processes in petrochemical industry applying fuzzy set theory," *Energy Cconversion and Mmanagement*, vol. 117, pp. 501–512, 2016.

[47] G. Ljubomir, D. Pamučar, S. Drobnjak, and H. R. Pourghasemi, "Modeling the spatial variability of forest fire susceptibility using geographical information systems and the analytical hierarchy process," in *Spatial Modeling in GIS and R for Earth and Environmental Sciences*, pp. 337–369, Elsevier, Amsterdam, Netherlands, 2019.

[48] P. Eachempati and P. R. Srivastava, "Personalized eElective sSelection: aA mMulti-cCriteria pPerspective in aAcademia," *International Journal of Strategic Decision Sciences (IJSDS)*, vol. 10, no. 4, pp. 43–63, 2019.

[49] T. Singh, A. Patnaik, G. Fekete, R. Chauhan, and B. Gangil, "Application of hybrid analytical hierarchy process and complex proportional assessment approach for optimal design of brake friction materials," *Polymer Composites*, vol. 40, no. 4, pp. 1602–1608, 2019.

[50] Y. Wind and T. L. Saaty, "Marketing applications of the analytic hierarchy process," *Management Science*, vol. 26, no. 7, pp. 641–658, 1980.

[51] M. B. Ayhan, "A ffuzzy AHP aapproach for ssupplier sselection pproblem: a ccase sstudy in a gear mmotor ccompany," 2013, https://arxiv.org/abs/1311.2886 arXiv preprint arXiv:1311.2886.

[52] O. Gogus and T. O. Boucher, "Strong transitivity, rationality and weak monotonicity in fuzzy pairwise comparisons," *Fuzzy Sets and Systems*, vol. 94, no. 1, pp. 133–144, 1998.

[53] A. Calik, "A multi-criteria evaluation for sustainable supplier selection based on fuzzy sets," *Business and Economics Research Journal*, vol. 10, no. 1, pp. 95–113, 2020.

[54] F. Cavallaro, E. K. Zavadskas, D. Streimikiene, and A. Mardani, "Assessment of concentrated solar power (CSP) technologies based on a modified intuitionistic fuzzy TOPSIS and trigonometric entropy weights," *Technological Forecasting and Social Change*, vol. 140, pp. 258–270, 20198.

[55] Y. Wind and T. L. Saaty, "Marketing applications of the analytic hierarchy process," *Management Science*, vol. 26, no. 7, pp. 641–658, 1980.

[56] S. Zhang, H. Gao, G. Wei, Y. Wei, and C. Wei, "Evaluation based on distance from average solution method for multiple criteria group decision making under picture 2-tuple linguistic environment," *Mathematics*, vol. 7, no. 3, p. 243, 2019.

[57] Ž. Stević, M. Vasiljević, A. Puška, I. Tanackov, R. Junevičius, and S. Vesković, "Evaluation of suppliers under uncertainty: a multiphase approach based on fuzzy AHP and fuzzy EDAS," *Transport*, vol. 34, no. 1, pp. 52–66, 2019.

[58] B. E. Narkhede, R. Raut, B. Gardas, H. T. Luong, and M. Jha, "Selection and evaluation of third party logistics service provider (3PLSP) by using an interpretive ranking process (IRP)," *Benchmarking: An International Journal*, vol. 24, no. 6, pp. 1597–1648, 2017.

[59] D. Sushil, "Interpretive ranking process," *Global Journal of Flexible Systems Management*, vol. 10, no. 4, pp. 1–10, 2009.

[60] G. Choudhury, "Demystifying India's poverty line: here's everything you need to know," 2015, https://www.hindustantimes.com/business/demystifying-india-s-poverty-line-here-s-everything-you-need-to-know/story-43vy1sQ7LrCZuezTakDnkM.html.

[61] H. Devulapalli and G. Agrawal, "Mapping bus transit services in Hyderabad––an illustrative example of the use of open geospatial data," *Transportation Research Procedia*, vol. 25, pp. 4196–4206, 2017.

[62] T. S. Babu, K. V. A. Kumar, and P. S. C. Teja, "Implementation of sSpeed fFlow mModels for the aAnalysis of sStreet pParking in Hyderabad," *The International Journal of Analytical and Experimental Modal Analysis*, vol. 12, no. 1, pp. 1076–1088, 2020.

[63] R. T. C. Baski, "ac bbus ffares to gget ccheaper in Hyderabad. Telangana today," 2020, https://telanganatoday.com/rtc-ac-bus-fares-to-get-cheaper-in-hyderabad.

[64] "TNM staff Hyderabad metro sees record 4.6 llakh ppassengers on New Year's Eve. The News Minute," 2020, https://www.thenewsminute.com/article/hyderabad-metro-sees-record-46-lakh-passengers-new-years-eve-115183.

[65] U. S. Mahtani and C. P. Garg, "An analysis of key factors of financial distress in airline companies in India using fuzzy AHP framework," *Transportation Research Part A: Policy and Practice*, vol. 117, pp. 87–102, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6664281, 14 pages
https://doi.org/10.1155/2021/6664281

Research Article

A Crowd Counting Framework Combining with Crowd Location

Jin Zhang,[1] Sheng Chen,[1] Sen Tian,[2] Wenan Gong,[3] Guoshan Cai,[4] and Ying Wang[5]

[1]*College of Informatica Science and Engineering, Hunan Normal University, Changsha 410081, China*
[2]*College of Mathematics and Statistics, Hunan Normal University, Changsha 410081, China*
[3]*Changsha Transportation Information Center, Changsha 410016, China*
[4]*Changsha Tianxia Yida Information Technology Co., Ltd., Changsha 410221, China*
[5]*School of Humanities and Management, Hunan University of Chinese Medicine, Changsha 410208, China*

Correspondence should be addressed to Jin Zhang; jinzhang@hunnu.edu.cn

Received 21 December 2020; Revised 29 December 2020; Accepted 4 February 2021; Published 17 February 2021

Academic Editor: Chi-Hua Chen

In the past ten years, crowd detection and counting have been applied in many fields such as station crowd statistics, urban safety prevention, and people flow statistics. However, obtaining accurate positions and improving the performance of crowd counting in dense scenes still face challenges, and it is worthwhile devoting much effort to this. In this paper, a new framework is proposed to resolve the problem. The proposed framework includes two parts. The first part is a fully convolutional neural network (CNN) consisting of backend and upsampling. In the first part, backend uses the residual network (ResNet) to encode the features of the input picture, and upsampling uses the deconvolution layer to decode the feature information. The first part processes the input image, and the processed image is input to the second part. The second part is a peak confidence map (PCM), which is proposed based on an improvement over the density map (DM). Compared with DM, PCM can not only solve the problem of crowd counting but also accurately predict the location of the person. The experimental results on several datasets (Beijing-BRT, Mall, Shanghai Tech, and UCF_CC_50 datasets) show that the proposed framework can achieve higher crowd counting performance in dense scenarios and can accurately predict the location of crowds.

1. Introduction

The crowd counting methods are used in videos and pictures to predict the number of people. For example, it's beneficial, especially in case of an emergency, such as Corona Virus Disease 2019. Otherwise, it can also be used to perform similar tasks, such as vehicle counting and cell counting under a microscope. Like other computer vision tasks, crowd counting also faces enormous challenges in terms of occlusion, background interference, and image distortion.

Many excellent models and algorithms are proposed to solve these problems in crowd counting. The methods for solving crowd counting can be classified into two categories: traditional methods and methods based on convolutional neural network (CNN). The conventional methods focus on carefully designed features extraction algorithms to solve this problem. However, the conventional methods are difficult to handle dense scenes. Due to the good performance of deep learning in various fields in recent years, the problem of crowd counting is increasingly being solved by CNN. CNN-based methods are easy to use and have better performance.

Crowd counting methods based on CNN consist of two categories: DM-based methods and detection-based methods. The DM-based method [1] first uses a normalized Gaussian kernel to represent the number of people, then predicts the DM through the CNN, and finally sums the DM to obtain the number of people. The detection-based method is to detect the number and location of the crowd by training a crowd detector. Compared with the detection-based methods, the DM-based methods have more robust to highly occluded scenes [2]. However, the DM-based methods lead to the following problems [3]: (1) higher the proportion of false positives and (2) loss of crowd location information.

As the crowd density increases, it is particularly important to study methods for dense scenes. However, most of

the current research methods only focus on the design of the network structure and ignore the fundamental problem brought by DM: "location information loss." Location information and the number of people are complementary to each other. Therefore, a new crowd detection and counting framework is proposed to solve this problem.

Our main contributions are as follows.

We propose a new network structure called ResNet-DC. It uses the ResNet [4], which performs well on classification problems, as backbone. It uses the deconvolution layer as upsampling. It is compatible with other powerful network structures so that we can migrate other network structures, and the structure is applied to both DM and PCM.

We propose a new PCM that links the crowd counting problem with the crowd detection problem. In dense scenes, PCM shows better performance than DM in the same network.

2. Related Work

For crowd counting, many powerful methods and algorithms are proposed. This section briefly describes two different methods: traditional methods and CNN-based methods.

2.1. Traditional Methods. In traditional crowd detection and counting methods, Chan and Vasconcelos [5] and Ryan et al. [6] proposed a regression-based method that predicts the number of people by first separating the background and then extracting features from the foreground. Lin and Davis [7] and Wang and Wang [8] proposed a detection-based method, which uses two consecutive video frame sequences. Idrees et al. [9] proposed an approach based on a carefully designed set of features: HOG. With HOG, head detection, Fourier analysis, and points of interest are integrated to avoid the disadvantages of a single feature. In traditional research methods, most research work focuses on carefully designed features to solve this problem. However, these methods are challenging to handle dense scenes or the image severely disturbed by the background.

2.2. Methods Based on CNN. With the development and application of deep learning, more and more research work is currently using CNN to solve crowd counting problems. At present, deep learning has been applied in many fields, such as traffic sign recognition [10], vehicle speed estimation [11], object tracking [12], and bus arrival prediction [13]. Compared with carefully designed solutions for feature extraction, CNN based methods are easy to use and have outstanding performance. CNN-based methods consist of two categories: the DM-based methods and the detection-based methods.

In DM-based methods, Zhang et al. [14] proposed a strategy based on DM in a cross-scene scenario, which randomly crops the image, divides the obtained features into two subtasks, and gets DM and the number of people through full connection. Ding et al. [15] proposed the use of a deeply recursive network (DR-ResNet). Unlike the previous ResNet, the ResNet block in DR-ResNet is constructed in different convolution, batch normalization (BN) [16], and rectified linear unit (ReLU) [17] order and then add to the input to adapt to the scene changes. When processing video data, the CNN-based method will only consider each video frame separately and ignore the temporal correlation of adjacent frames. Xiong et al. [18] highlighted a new variant of CNN, called CNN LSTM, which captures space and time dependencies. To obtain high resolution DM, Liu et al. [19] proposed a method to optimize the multicolumn convolution neural network by learning global features and recover the lost details in downsampling by deconvolution. To adapt to the characteristics of multiscale crowds, Zhang et al. [1] first proposed a method to solve the scale problem through different convolution kernel sizes. Sam et al. [20] proposed the use of a switching convolutional neural network, which maps image patches to specific CNN columns. Sang et al. [21] optimized the geometric adaptive Gaussian kernel function of SaCNN to generate a higher quality real DM. Kong et al. [22] proposed an adaptive attention mechanism method to automatically adjust the network structure through the crowd size.

In the detection-based methods, [2, 23, 24] all use Faster R-CNN [25] as the crowd detector. To overcome the limitations of pedestrian detectors, Saqib et al. [23] proposed a motion-guided filter (MGF), which uses temporal and spatial information among successive frames of video to recover lost details. The performance of the detector in dense scenes is improved, but this scheme is only applicable to video stream data. In dense scenes, due to the severe occlusion, Vora [2] and Kong et al. [22] detected the crowd heads, which increased the accuracy of detection. Vora [2] proposed faster R-CNN directly for binary classification tasks, to determine whether the detection frame is a human head and to reduce the number of anchor boxes according to the human head scale, speeding up the detection process. Basalamah et al. [24] and others proposed a Faster R-CNN-based scale driven convolutional neural network (SD-CNN) model to detect crowd heads and to solve the problem of different head sizes in video streams based on a scale map.

3. A New Framework for Crowd Detection and Counting Combining RESNET-DC and PCM

The framework includes two parts. (1) The first part is a full CNN, namely, ResNet-DC, which consists of backend and upsampling. (2) The second part is PCM, which contains information about the location. In this section, the proposed framework is introduced firstly. Then, two critical parts of the framework are described in detail. Finally, some training details are shown.

3.1. Framework Structure. As shown in Figure 1, there are three steps in the structure of the proposed framework for crowd detection and counting. The first step aims to extract input image features based on a CNN consisting of backend and upsampling. Backend shown in Figure 2 uses the ResNet to extract the features, and upsampling shown in Figure 3

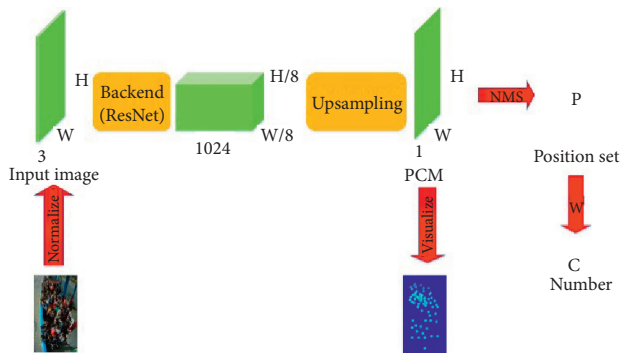


Figure 1: The structure of the proposed framework for crowd detection and counting.

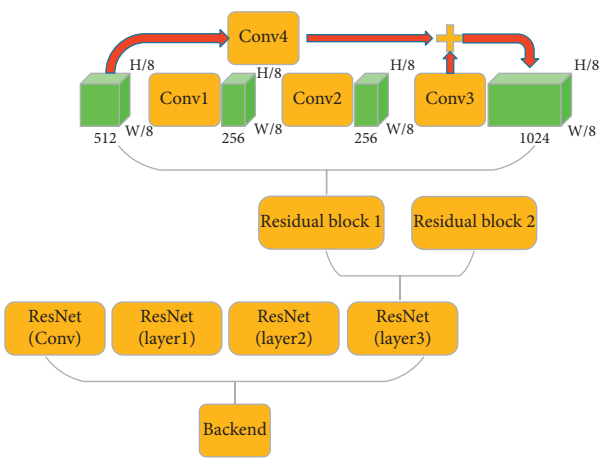


Figure 2: The network structure of backend.

uses the deconvolution layers to restore the feature map scale. The second step aims to predict high-quality PCM. The last step is to analyze the estimated position set P to get the number of people and location. To obtain the location information of the crowd, it is only necessary to perform nonmaximum suppression on PCM to get the location set. Therefore, we only need to count the location of the crowd to get the number of people.

3.2. ResNet-DC. The first part of the proposed framework is named as ResNet-DC. In ResNet-DC, backend extracts the features of the input image and reduces the input size by eight times, and upsampling restores the size of the feature map to obtain a high-quality PCM.

3.2.1. Backend. In this work, ResNet-18 [4] is used as the backbone network, which has outstanding performance in classification problems. In the backbone network, the deeper the network, the more increased the memory, training, and inference time. Due to the real-time nature of crowd detection, it is reasonable to use the first to third layers of ResNet. As the step size increases, the downsampling of the feature map increases. The step size of the residual block 1 in the third layer of ResNet is changed from two to one according to the crowd counting framework [26] to avoid severe loss of location information due to downsampling. Figure 2 shows the modified structure of the first residual block in layer three of ResNet. The detailed configuration is shown in Table 1. The subsequent residual blocks still retain the original design of ResNet. Under this setting, backend extracts the feature information of the original image and performs downsampling to obtain a feature map that is eight times smaller than the original.

3.2.2. Upsampling. In crowded scenes, excessive downsampling causes loss of feature information (especially location information). It is a feasible method to use the deconvolution layer to recover the feature information and obtain high-quality PCM. Deconvolution can be regarded as the inverse process of convolution and pooling. Long et al. [27] show that the deconvolution layer can recover more feature information than using convolution and bilinear interpolation. In this paper, the structure of upsampling is shown in Figure 3. It consists of two and three deconvolution layers. The first convolutional layer is responsible for compressing the channels of the feature map. The three deconvolution layers in the middle are accountable for upsampling the feature map to the original image size. The last convolutional layer is responsible for mapping the feature map to PCM. Table 2 shows configuration information for upsampling.

$$G_{\sigma_i}x, y = \begin{cases} \alpha e^{-\left((x-x_i)^2+(y-y_i)^2/2\sigma^2\right)}, & x_i, y_i - \dfrac{ksize}{2} \le x, \\ & y \le x_i, y_i + \dfrac{ksize}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

Under the above structure, ResNet-DC can restore the feature map reduced by backbone to the same size as the input. In this way, the predicted feature map will not ignore some peaks due to overlapping peaks.

3.3. Peak Confidence Map. PCM, an improvement over DM, is designed and compares with DM in this section. Then, a nonmaximum suppression algorithm is introduced to obtain crowd information from PCM.

3.3.1. Density Map. The density map design is based on [1, 28]. For a head position (x_i, y_i) in an image, a normalized Gaussian kernel function $G_{\sigma_i}x, y$ is generated in the neighborhood of its $ksize \times ksize$. $G_{\sigma_i}x, y$ can be expressed as follows: where α is the normalization factor so that $\sum G_{\sigma_i}x, y = 1$. σ_i is the variance of the Gaussian kernel of the ith head. In traditional DM, it is designed as a constant. To convert the marked points into a density function, the

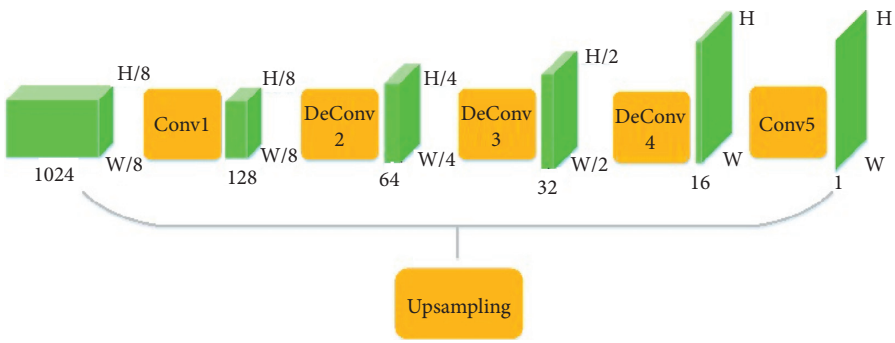


Figure 3: The structure of upsampling.

Table 1: The configuration of modified residual block 1 in ResNet18.

Layers name	Kernel	Stride	Input padding	Bias	BN	ReLU
Conv1	$256 * 1 * 1 * 512$	1	0	False	Yes	No
Conv2	$256 * 3 * 3 * 256$	1	1	False	Yes	No
Conv3	$256 * 1 * 1 * 256$	1	0	False	Yes	No
Conv4	$256 * 1 * 1 * 1024$	1	0	False	Yes	Yes

Table 2: The configuration of upsampling layer.

Layers name	Kernel	Stride	Input/output padding	Bias	BN	ReLU
Conv1	$128 * 3 * 3 * 1024$	1	1/-	Yes	Yes	Yes
DeConv2	$64 * 3 * 3 * 128$	2	1/1	Yes	Yes	Yes
DeConv3	$32 * 3 * 3 * 64$	2	1/1	Yes	Yes	Yes
DeConv4	$16 * 3 * 3 * 32$	2	1/1	Yes	Yes	Yes
Conv5	$1 * 1 * 1 * 16$	1	0/-	Yes	No	Yes

normalized Gaussian kernel function G_{σ_i} at different positions needs to be summed. The density function $F(x, y)$ can be expressed as follows:

$$\begin{aligned} M(x_i, y_i) &= \max\left\{G_{\delta_i}(x, y), M(x_{i-1}, y_{i-1})\right\}, \\ F(x, y) &= M(x_N, y_N). \end{aligned} \tag{2}$$

where $M(x_i, y_i)$ represents a density function that already contains i head positions and N represents the number of people in the ith image.

However, each head position is sampled in a 3D scene. Due to perspective distortion, different head sizes in the image are caused. Zhang et al. [1] found that the denser the crowd, the smaller the head size. To solve the problem of perspective distortion, [1] proposed a DM using an adaptive geometric Gaussian kernel based on the previous findings; that is, $\delta_i = \beta \bar{d}^i$. $\bar{d}^i$ represents the average of the distance between the ith head position and the k nearest heads, and $\beta = 0.3$ is obtained through experiments. Since $G_{\delta_i}(x, y)$ is normalized, each position corresponds to a Gaussian kernel function or adaptive geometric Gaussian kernel function with a sum of 1. By summing the pixels of the density function $F(x, y)$, the number of people can be obtained. However, due to the addition operation, false peaks may occur, which leads to the loss of position information. For example, there is a situation as shown in the left of Figure 4 (represented in one dimension), and the red and blue curves represent the Gaussian kernel function that transforms the position information of different people. It is easy to know that $x1$ and $x3$ represent different head positions, and the black curve can be obtained after the addition. Since a false peak $x2$ is generated at this time, it is impossible to determine which peak is the head position.

3.3.2. Peak Confidence Map. The different Gaussian kernel peaks correspond to the marked position of the head. In this paper, a design scheme for PCM that overcomes the shortcomings of location information loss is proposed. Unlike previous DM, the peak confidence function performs a maximum operation. PCM is defined in this paper as follows:

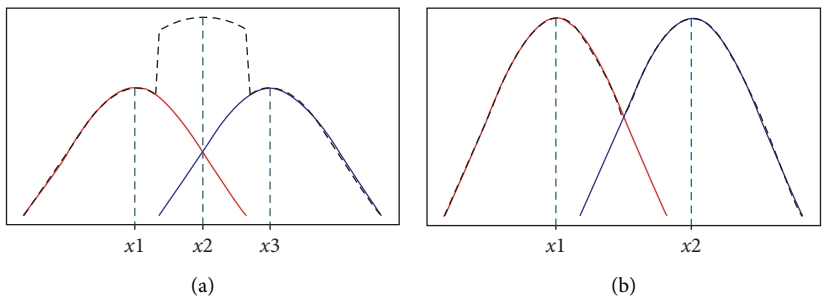


Figure 4: The 1D DM and PCM.

$$G_{\sigma_i}x, y = \begin{cases} e^{-\left((x-x_i)^2+(y-y_i)^2/2\sigma^2\right)}, & x_i, y_i - \dfrac{ksize}{2}, \leq x, \\ & y \leq x_i, y_i + \dfrac{ksize}{2}, \\ 0, & \text{otherwise}, \end{cases}$$

$$M(x_i, y_i) = \max\left\{G_{\delta_i}(x, y), M(x_{i-1}, y_{i-1})\right\},$$

$$F(x, y) = M(x_N, y_N), \tag{3}$$

where $G_{\delta_i}(x, y)$ represents the Gaussian kernel corresponding to the ith head position, $M(x_i, y_i)$ represents a confidence function that already includes *i-head* positions, N represents the number of persons in the image, and σ_i is the ith heads correspond to the variance of the Gaussian kernel. Compared with DM, PCM no longer normalizes the Gaussian kernel because it uses the number of peaks to count the number of people and does not need to be summed like DM. The reason for named PCM is that (1) the peak represents the number and location of the crowd and (2) the closer to the head position, the higher the value. To some extent, it can reflect the confidence that a certain head position exists in PCM. Figure 4 shows the difference between PCM and DM. As shown to the right of Figure 4, if it is expressed in one dimension, the red and blue curves in the figure represent the Gaussian kernel functions corresponding to different heads positions. The black curve shows the results obtained by taking the maximum of different Gaussian kernel functions. As can be seen from the black curve, the two peaks exactly represent the head positions of different people. During the experiment, the peak confidence function was regressed to make the network produce different peaks at different people's head positions. By obtaining the extreme point from PCM to get the position of the peak, it is easy to know how many people will produce how many peaks.

According to the design method of PCM and DM, PCM and DM on Beijing-BRT [15], Mall [29], Shanghai Tech [1], and UCF_CC_50 [9] can be calculated. Figure 5 shows that there is not much difference between PCM and DM when the crowd is scattered. When the crowd is dense, the maximum value of PCM is at the head position of each person, and the location information and the crowd distribution can be calculated more precisely. But in DM, the denser the crowd, the greater the value, so the position information is lost.

In general, PCM and DM have the following differences. (1) DM takes the sum between Gaussian kernels, while PCM takes the maximum value between Gaussian kernels. (2) DM needs to normalize the Gaussian kernel, but PCM does not. (3) DM calculates the number of people by calculating the sum, and PCM calculates the position and the number of people by calculating the peak value.

3.3.3. Nonmaximum Suppression. Nonmaximum suppression aims at maximum local searching, that is, finding extreme points. In DM, due to the interference of false peaks, many incorrect positions will be detected by nonmaximum suppression method. So, it uses the regularized Gaussian kernel to calculate the number of people. This leads to the loss of location information. But in PCM, since each person's head corresponds to a peak, nonmaximum suppression becomes possible. The extreme point set P is calculated as follows:

$$P = \bigcup_{i=1}^{W} \bigcup_{j=1}^{H} \left\{\operatorname{argmax}\left(F\left(x_i, y_j\right), \delta_4\right) > \vartheta\right\}, \tag{4}$$

where $F(x_i, y_j)$ denotes the (i, j)th pixel in PCM with the size of (W, H), δ_4 represents the four neighborhoods of pixels, ϑ is the confidence, and argmax denotes the subscript to get the maximum value. For each pixel of PCM, (7) compares it with its four domains. If the point is the maximum in four domains, then the pixel is the local maximum, that is, the extreme point. In other words, the head position P is a set: it is a local maximum and greater than the confidence.

3.4. Train Details. This section gives detailed training information on ResNet-DC. By using pretrained ResNet, ResNet-DC can quickly converge.

3.4.1. Label Normalize. The current work in [26] points out that a regression value will affect the performance of the network if a regression value is too small in DM. Considering the same effect on PCM, we multiply PCM by a factor of amplification. In this paper, we set the amplification factor

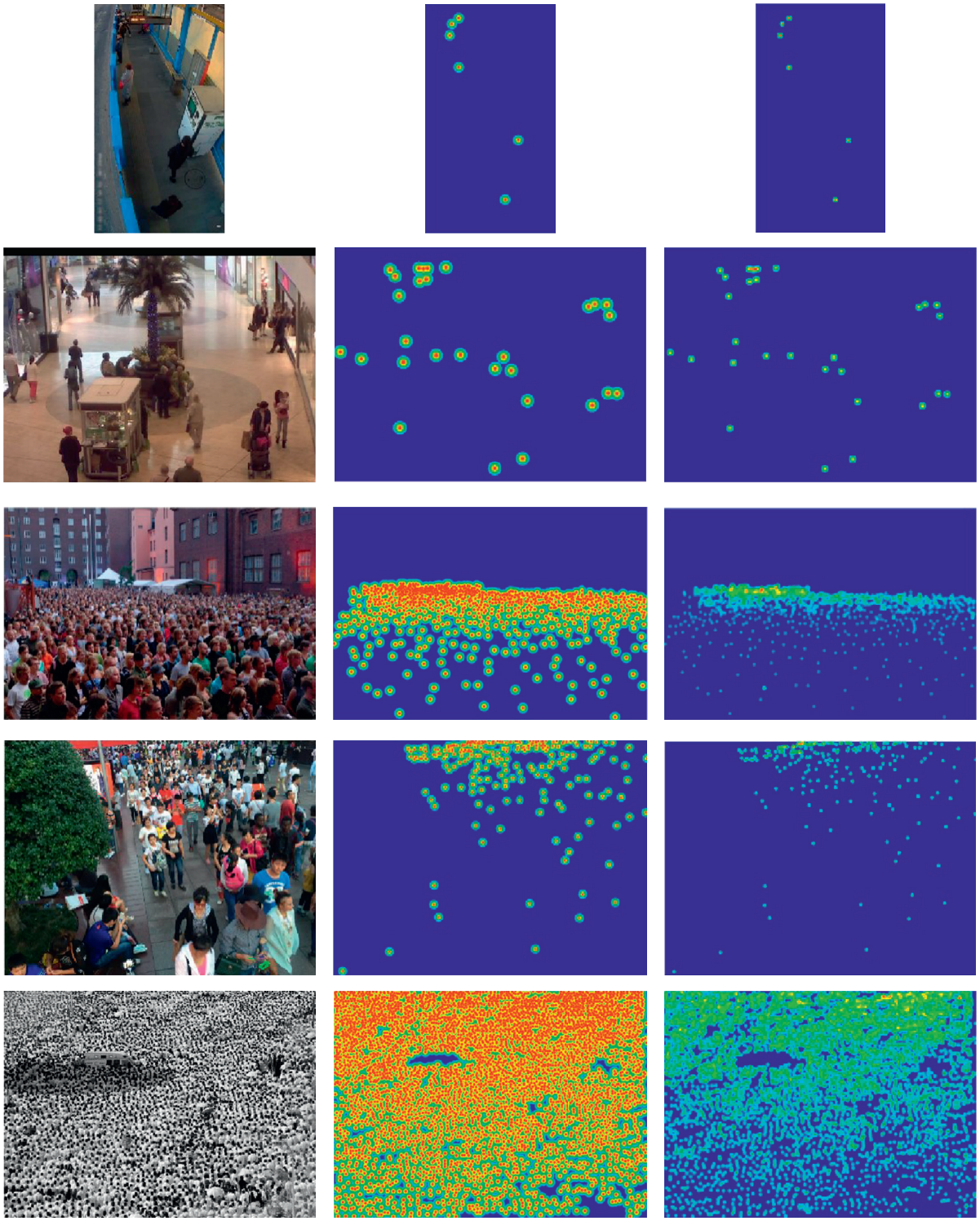


Figure 5: Comparison of density maps and peak confidence maps, sampled from Beijing-BRT (1st row), Mall (2nd row), Shanghai Tech Part A (3rd row), Shanghai Tech Part B (4th row), and UCF_CC_50 datasets (5th and 6th rows). The original pictures in the first column are sampled from different datasets. The picture in the second column represents the corresponding peak confidence maps. The picture in the third column represents the related density maps.

to 10. The reason for setting the magnification factor is that if the value of the PCM is too small, the network is easy to predict the wrong peak value, which is caused by the small difference between adjacent values. If the value of the PCM is too large, it is difficult for the network to converge, which is caused by the excessively large loss value.

3.4.2. Data Augment. The current work in [1] obtains nine times images by cropping at different positions. Since cropping may cause the loss of global information, in our experiments, we only flip the original image horizontally to obtain twice the image.

3.4.3. Loss Function. Most research work [1, 15, 20] uses the mean square loss to evaluate the error. In this paper, the mean square loss is also used. The MSE loss function is defined as follows:

$$L_{mse}(\theta) = \frac{1}{2N}\left|F(I_i;\ \theta) - G_i\right|^2, \tag{5}$$

where θ represents the parameters that ResNet-DC needs to learn, N represents the number of pictures, $F(I_i;\theta)$ represents PCM predicted by the ith input image I, and G_i represents the ground-truth PCM of the ith input image I. But when the mean square loss is only used, the network is biased towards more peaks predicted. Although the mean square loss can penalize the error between the ground-truth PCM and the estimated PCM, it ignores the relationship between adjacent pixels. Compared with DM, PCM has a stricter relationship with neighboring pixels. The reason for the extra peak is caused by ignoring the relationship between adjacent pixels.

In PCM, considering the importance of the relationship between adjacent pixels, a feasible solution is to calculate the difference between adjacent pixels. As we all know, the relationship between adjacent pixels can express important information. For example, the pixel values that are close to each other represent the same element, and the pixel values that are relatively different represent the boundaries of different elements. In order to express the above information, we use a convolution kernel with *kernel* = [[−1, −1, −1], [−1, 9, −1], [−1, -1, −1]]. The specific convolution kernel form is not important. We can use *kernel* = [[0, −1, 0], [−1, 5, −1], [0, −1, 0]] to achieve the same effect. Only the previous convolution kernel takes into account the values of the four corners. In this work, we use a convolution *kernel* = [[−1, −1, −1], [−1, 9, −1], [−1, −1, −1]] of size 3×3 to convolve with PCM to get the relationship between adjacent pixels. The loss is defined as follows:

$$L_{\ker}(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left|F(I_i;\ \theta) * \text{kernel} - G_i * \text{kernel}\right|^2, \tag{6}$$

We use the kernel to convolve with PCM to obtain the difference value between the center point and its eight neighborhoods and then calculate the mean square error within the area. The total loss $L(\theta)$ can be calculated as follows:

$$L(\theta) = L_{mse}(\theta) + L_{\ker}(\theta). \tag{7}$$

3.4.4. Learning Setting. According to transfer learning in [30] to accelerate model convergence, a straightforward way to train the ResNet-DC is used as an end-to-end structure. Backend is fine-tuned from a well-trained ResNet-18 [4]. For upsampling, the initial values come from a Gaussian initialization with 0.01 standard deviation. Using the Adam optimization algorithm, the learning rate is $5e-5$, and the weight decay rate is $1e-4$. The input image is regularized (mean and variance on the Imagenet dataset) and then trained on the dataset to predict PCM. At the same time, each iteration on the training set is verified on the validation set, and the best model in the validation set is retained.

4. Performance Evaluation

In this section, several datasets are used to evaluate performance. The crowd count evaluation metric and location evaluation metric are proposed. Based on the datasets and the metrics, the performance of different methods is compared and analyzed.

4.1. Dataset. Currently, the mainstream crowd count dataset includes Beijing-BRT [15], Mall [29], Shanghai Tech [1], and UCF_CC_50 [9]. In the framework, we performed experiments on the above four datasets, each of which is described as Table 3. In Beijing-BRT, we divided the training set and test set according to the criteria of [15]. In Mall, we divide the training set and test set according to the criteria of [18]. In Shanghai Tech, we divide the training set and test set according to the criteria of [14]. In UCF_CC_50, we use 5-fold cross-validation according to the standard of [9]. In these datasets, due to the different image resolutions of the Shanghai Part A and UCF_CC_50 datasets, we counted their average resolutions. We resized the image size so that it is closest to the average resolution and divisible by eight.

4.2. Evaluation Metric. According to the existing methods [1, 14], the mean square absolute error (MAE) and mean squared error (MSE) are used to evaluate the performance of crowd counting, which are defined as follows:

$$\begin{aligned} \text{MAE} &= \frac{1}{N}\sum_{i=1}^{N}\left|c_i - \hat{c}_i\right|, \\ \text{MSE} &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(c_i - \hat{c}_i\right)^2}, \end{aligned} \tag{8}$$

where N is the number of pictures, c_i is the number of people in the ith picture, and $\hat{c}_i$ is the number of people predicted in the ith picture. To some extent, the mean square absolute error can be regarded as the accuracy of the prediction, and the mean square average error can be regarded as the generalization ability of the model. These two indicators are equally important. From the value of MAE and MSE, the lower the value, the higher the accuracy.

To quantitatively analyze the position performance, we use a method similar to object detection to evaluate the position performance as follows. (1) If a real position of the $S \times S$ neighborhood exists in the predicted position, we classify it as true positive. (2) If a predicted position does not belong to any of the real positions of the $S \times S$ neighborhood,

Table 3: Summary of the four datasets.

Datasets		Images	Count	Avg. density	Resolution	Avg. resolution	Resize
Beijing-BRT		1280	16,795	13.1	320 ∗ 640	—	—
Mall		2000	62,325	31.2	640 × 480	—	—
Shanghai tech	Part A	482	241,677	501.4	Different	868 ∗ 589	872 ∗ 592
	Part B	716	88,488	123.6	1024 × 768	—	—
UCF_CC_50		50	63,974	1279.5	Different	902 ∗ 653	904 ∗ 656

we classify it as false positive. Then, the standard Average Precision (AP) and Average Recall (AR) scores are calculated. In this experiment, S represents the allowed position error. We believe that due to the differences in manual marking, not all positions are accurately marked in the center of the human head, and there will be some errors. Therefore, when S is set to eight, it is reasonable to predict the position as the true positive.

4.3. Experimental Results and Analysis. In the experiment, we use the framework proposed in this paper to solve the crowd counting problem and crowd location prediction simultaneously. The experimental results on the above four datasets show that the proposed framework is not only suitable for dense scenes but also can predict the position of the crowd.

4.3.1. Counting Performance. In the experiment, we compared the crowd counting performance of DM and PCM. At the same time, we also compared it with other powerful algorithms. Tables 4–7 show the performance results of crowd counting on four different data sets. In DM, we use ResNet-DC to compare with other excellent algorithms, and the results show that the ResNet-DC has made slight progress in the Shanghai Tech part A (0.2 MAE) dataset and achieved good performance on other datasets. The results are acceptable because we used the simplest ResNet-18 as backend network. We can also use other deeper networks such as ResNet-32, ResNet-50, and ResNet-101. When we use PCM in ResNet-DC, we have performed excellent performance in Shanghai Tech Part A (2.33 MAE, 6.8 MSE) and good performance on other datasets.

4.3.2. Localization Performance. Because there are fewer experiments on localization on the crowd counting dataset, we only compare the AP and AR of different methods on the UCF_CC_50 dataset, as shown in Table 8. Compared with the current best algorithm SD-CNN [24], our approach is slightly worse on AP. But we have reached the best level in AR, and the improvement of 1.48 AP is better than the current best algorithm. We believe that this is because we only place the position with higher confidence (confidence is 0.5) as the location. Higher confidence leads to higher AP, but also lower AR. And because AP is soft, this leads to the degradation of MAE performance. Table 9 shows the position performance of our algorithm on the other three datasets. We found that although the performance of AP and AR gradually decreased with the increase of the crowd density, even on the worst-performing Shanghai Tech Part

Table 4: Counting performance of the different methods on Beijing-BRT.

Methods	Type	Position	Beijing-BRT	
			MAE	MSE
MCNN [1]	DM	No	2.24	3.35
FCNCC [31]	DM	No	1.74	2.43
ResNet-14 [15]	DM	No	1.48	2.22
DR-ResNet [15]	DM	No	1.39	**2.00**
ResNet-DC	DM	No	**1.36**	2.02
ResNet-DC	PCM	Yes	1.40	2.16

Table 5: Counting performance of the different methods on Mall.

Methods	Type	Position	Mall	
			MAE	MSE
CNNLSTM [18]	DM	No	2.24	8.5
ASA [22]	DM	No	2.3	3.0
MGF [23]	Detection	**Yes**	**1.89**	7.29
ResNet-DC	DM	No	2.33	**2.89**
ResNet-DC	PCM	Yes	2.49	3.14

Table 6: Counting performance of the different methods on Shanghai Tech.

Methods	Type	Position	Shanghai tech			
			Part A		Part B	
			MAE	MSE	MAE	MSE
MCNN [1]	DM	No	110.2	173.2	26.4	41.3
Switching CNN [20]	DM	No	90.4	135.0	20.0	33.4
MNCS [19]	DM	No	86.6	129.7	19.3	35.3
ASA [22]	DM	No	83.9	133.3	18.6	31.1
Sang et al. [21]	DM	No	75.84	124.9	11.0	**18.6**
ResNet-DC	DM	No	79.85	131.2	**10.8**	**18.6**
ResNet-DC	PCM	**Yes**	**73.51**	**118.1**	13.3	22.5

Table 7: Counting performance of the different methods on UCF_CC_50.

Methods	Type	Position	UCF_CC_50	
			MAE	MSE
Faster R-CNN [25]	Detection	**Yes**	592.09	672.19
MCNN [1]	DM	No	377.6	509.1
Switching CNN [20]	DM	No	318.1	439.2
MNCS [19]	DM	No	306.7	396.3
DA-Net [32]	DM	No	290.8	326.5
SD-CNN [24]	Detection	**Yes**	**235.74**	345.6
ResNet-DC	DM	No	286.3	415.0
ResNet-DC	PCM	**Yes**	254.78	**326.16**

Table 8: Localization performance of the different methods on UCF_CC_50.

Method	Type	UCF-CC-50	
		AP%	AR%
Faster R-CNN [25]	Detection	14.52	12.69
Kang et al. [3]	Detection	24.13	30.27
SD-CNN [24]	Detection	**45.67**	40.12
ResNet-DC	PCM	43.8	**41.6**

Table 9: Localization performance of ResNet-DC and PCM on Beijing-BRT, Mall, and Shanghai Tech.

Datasets		ResNet-DC + PCM	
		AP%	AR%
Beijing-BRT		65	66
Mall		66	63
Shanghai tech	Part A	59	59
	Part B	61	63

A, both AP and AR reached 59%. The result of four datasets shows that our algorithm can detect reliable locations even in dense scenes.

4.3.3. Result Analysis. Why is the performance of DM slightly better than PCM in sparse scenarios? Under the same network structure, the results show that in the sparse crowd scene (Beijing-BRT, Mall, and Shanghai Tech Part B), the design of DM is slightly better than PCM for crowd counting. We consider that this is since (5) has robustness for DM. In DM, it ignores the relationship between adjacent pixels. Even if there is a small amount of value prediction error, the impact on the crowd count is relatively small. PCM pays more attention to the comparison between adjacent pixels. Although (6) can mitigate the error value, it has not reached the optimal performance. Besides, we visualized the prediction results of PCM and DM under the same network structure. As shown in Figure 6, due to picture distortion, ResNet-DC loses information about people in the distance. Affected by the shooting environment, ResNet-DC lost the information of nearby people. For the missing information, PCM shows a lower confidence (lower than the confidence value 0.5), so PCM directly discards these values. Instead, the DM will add these values to the number of people. As a result, the predicted number of DM is closer to the true value than PCM.

Why is the performance of PCM better than DM in dense scenarios? Under the same network structure, the results show that in crowded scenes (Shanghai Tech Part A and UCF_CC_50), the crowd counting performance of PCM is significantly improved compared to the DM method. We also visualized the prediction results of PCM and DM under the same network structure. As shown in Figure 7, due to the defects of the convolutional network, the predicted picture in the dense scene is disturbed by the background (red rectangle). In PCM, since we define the peak value to be greater than the threshold value, nonmaximum suppression can filter out small activation values. In DM, these interference values are usually added to the number of people, resulting in the DM.

The method predicts a larger number of people. Besides, the results show that the network is generally interfered by occlusion in dense scenes, resulting in incorrect predictions (black rectangles) in dense areas. Because PCM combines position information, it is only sensitive to peaks. DM will directly add these false values to the number of people, which leads to the instability of the forecast results.

Why is PCM better than DM? First of all, due to design differences, PCM naturally contains location information, but DM does not. Secondly, since the peak value indicates the location of the crowd, PCM can ignore small activation values, thereby significantly reducing background interference. Conversely, DM adds these interference values to the crowd count. Finally, because PCM focuses on local maximums, it can ignore the second largest activation values generated in crowded places. Conversely, DM will also add the false activation values to the crowd count. We also visualized some of the results on the test set in Figure 8. Figure 8 shows that the predicted PCM generally has a high confidence level for the predicted crowd location on a dataset with low crowd density (Beijing-BRT, Mall, and Shanghai Tech Part B). On a dense crowd dataset (Shanghai Tech Part A and UCF_CC_50), the confidence level of the predicted PCM for the predicted crowd location is generally lower. As the crowd density increases, the peak confidence decreases. This phenomenon is consistent with people's intuitive feelings. At the same time, Figure 8 also shows that PCM can accurately predict the location of the crowd, which DM cannot do.

In general, PCM shows better performance than DM when faced with computer vision occlusion, background interference, and image distortion. Specifically, for occlusion and image distortion issues, PCM only considers the peak value. That is to say, even if there are overlapping or different-sized headers, PCM only needs to consider whether there is otherwise in the prediction result and does not need to consider the global information of the headers like DM. As for background interference, PCM can also filter out the interference information.

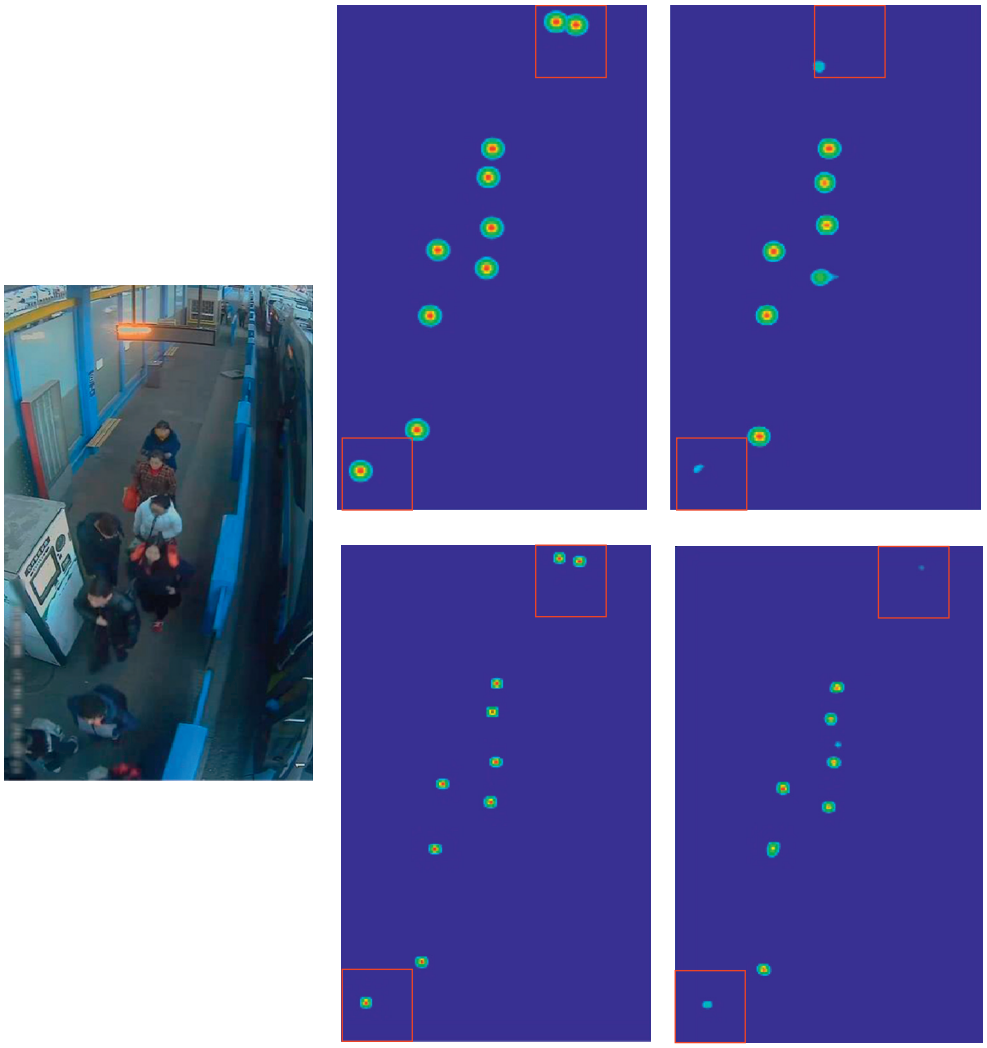


Figure 6: In the sparse scene, the comparison of predicted DM and PCM on ResNet-DC. The images sample from Beijing-BRT. The second column represents the true PCM or DM. The third column represents the predicted result. The first row represents PCM, and the second row represents DM.

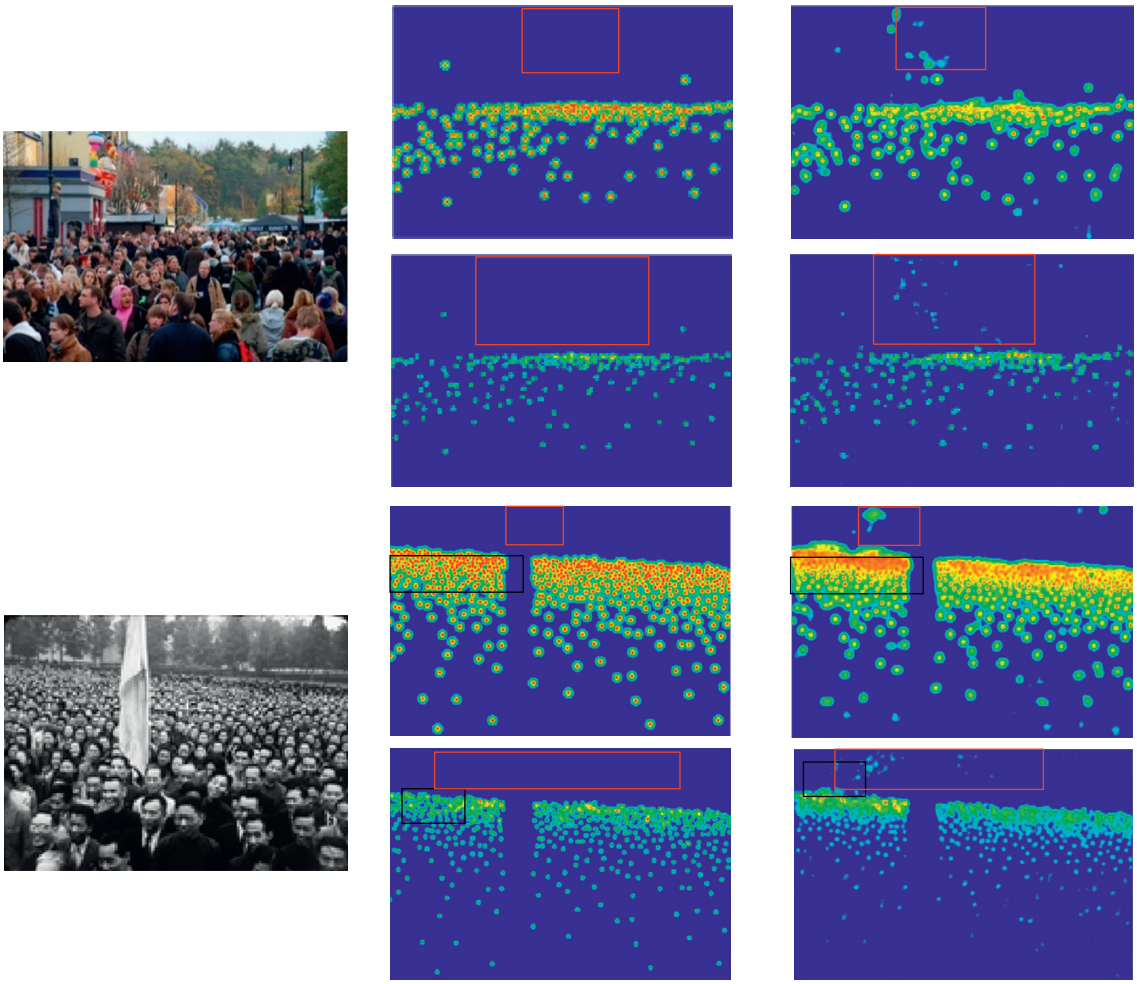


Figure 7: In dense scenarios, the comparison of predicted DM and PCM on ResNet-DC. The images sample from Shanghai Part A The 2nd column represents true PCM or DM. The 3rd column represents the predicted result. The 1st and 3rd rows represent PCM, and the 2nd and 4th rows represent DM.

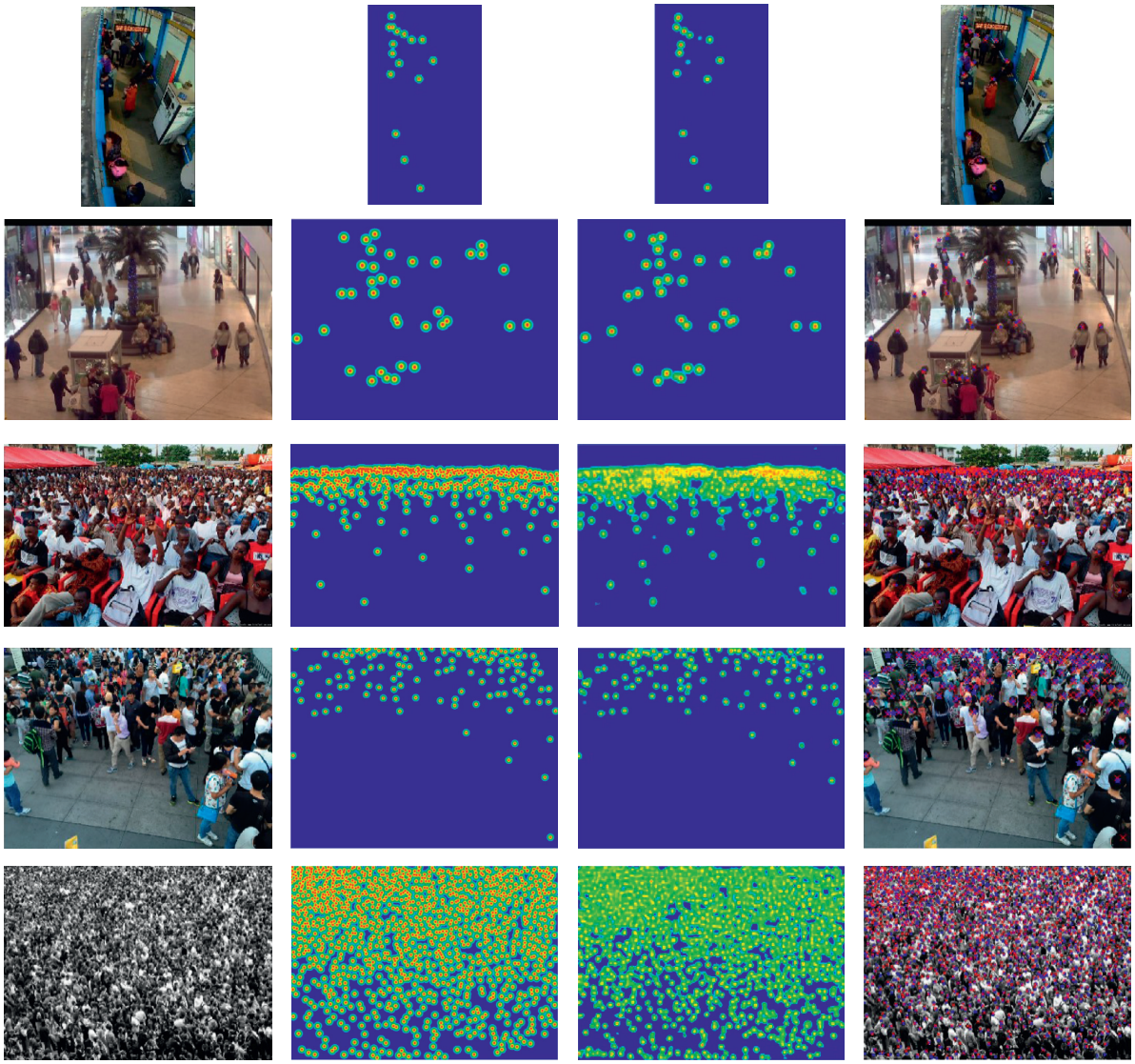


Figure 8: Results of sample images from Beijing-BRT (1st row), Mall (2nd row), Shanghai Part A (3rd row), Shanghai Part B (4th row), and UCF_CC_50 (5th row). The original pictures in the first column are sampled from different datasets. The pictures in the second column represent the corresponding ground-truth peak confidence map. The pictures in the third column represent the corresponding estimated peak confidence map. The pictures in the fourth column represents the ground-truth (indicating in red) and estimated (indicating in blue) position.

5. Conclusion

In this paper, a new framework is proposed to solve the problem of crowding detection and counting at the same time. The framework combines ResNet-DC with PCM to predict the number of people and the position of the person. ResNet-DC is a full CNN consisting of backend and upsampling. Backend is used as a feature extractor, and upsampling maps the extracted features into a high-quality PCM. The entire network is an end-to-end structure, and it is easy to migrate other excellent models to ResNet-DC. PCM retains the crowd distribution and location information. It can obtain position information through nonmaximum suppression and is also an effective method to solve background interference. Experimental results on four public datasets show that the proposed framework has good crowd counting performance and can even get accurate location information.

Data Availability

The related codes and data in the literature are released at https://github.com/Yuesheng321/RestNet-DC.git.

Conflicts of Interest

The authors declare that there are no conflicts of interest in the submission of this manuscript.

Authors' Contributions

The manuscript was approved by all authors for publication.

Acknowledgments

This work was supported by the education and research projects of Hunan Provincial Education Department (JG2018A012, XiangJiaoTong [2019] no. 291-410, no. 248-27, no. 370, [2020], no. 9, no. 90, and no. 233 HNKCSZ-2020-0122), the projects of the Ministry of Education of the People's Republic of China (201901051021), and the Science and Technology Progress and Innovation Project of Hunan Provincial Department of Transportation (no. 201927).

References

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas*, pp. 589–597, NV, USA, June 2016.

[2] A. Vora, "FCHD: a fast and accurate head detector," 2018, https://arxiv.org/abs/1809.08766.

[3] D. Kang, Z. Ma, and A. B. Chan, "Beyond counting: comparisons of density maps for crowd analysis task—counting, detection, and tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 5, pp. 1408–1422, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[5] A. B. Chan and N. Vasconcelos, "Bayesian Poisson regression for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 545–551, New York; NY, USA, September 2009.

[6] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression models," *Computer Vision and Image Understanding*, vol. 130, pp. 1–17, 2015.

[7] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 606–618, 2010.

[8] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3401–3408, Colorado Springs, CO, USA, June 2011.

[9] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2547–2554, Portland, OR, USA, June 2013.

[10] J. Zhang, Z. Xie, J. Sun, X. Zou, and J. Wang, "A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection," *IEEE Access*, vol. 8, pp. 29742–29754, 2020.

[11] C.-H. Chen, "A cell probe-based method for vehicle speed estimation," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 1, pp. 265–267, 2020.

[12] J. M. Zhang, J. Sun, J. Wang, and X. G. Yue, "Visual object tracking based on residual network and cascaded correlation filters," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 1–14, 2020.

[13] C.-H. Chen, "An arrival time prediction method for bus system," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4231-4232, 2018.

[14] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–841, Boston, MA, USA, June 2015.

[15] X. Ding, Z. Lin, F. He, Y. Wang, and Y. Huang, "A deeply-recursive convolutional network for crowd counting," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1942–1946, Calgary, AB, Canada, April 2018.

[16] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," 2010, https://www.cs.toronto.edu/%7Efritz/absps/reluICML.pdf.

[18] F. Xiong, X. Shi, and DY. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *Proceedings of the International Computer Vision (ICCV)*, pp. 1861–1870, Venice, Italy, October 2017.

[19] Z. Liu, Y. Chen, B. Chen, L. Zhu, D. Wu, and G. Shen, "Crowd counting method based on convolutional neural network with global density feature," *IEEE Access*, vol. 7, pp. 88789–88798, 2019.

[20] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings fo the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4031–4039, Honolulu, Hl, USA, July 2017.

[21] J. Sang, W. Wu, H. Luo et al., "Improved crowd counting method based on scale-adaptive convolutional neural network," *IEEE Access*, vol. 7, pp. 24411–24419, 2019.

[22] W. Kong, H. Li, G. Xing, and F. Zhao, "An automatic scale-adaptive approach with attention mechanism-based crowd spatial information for crowd counting," *IEEE Access*, vol. 7, pp. 66215–66225, 2019.

[23] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein, "Crowd counting in low-resolution crowded scenes using region-based deep convolutional neural networks," *IEEE Access*, vol. 7, pp. 35317–35329, 2019.

[24] S. Basalamah, S. D. Khan, and H. Ullah, "Scale driven convolution neural network model for people counting and localization in crowd scenes," *IEEE ACCESS*, vol. 7, pp. 71576–71584, 2019.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection proposal networks," *Advances in Neural Information Processing Systems*, vol. 39, no. 6, pp. 91–99, 2015.

[26] J. Gao W, B. Zhao, D. Wang, C. Gao, and J. Wen, C3 framework: an open-source pytorch code for crowd counting," 2019, https://arxiv.org/abs/1907.02724.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, June 2015.

[28] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proceedings of the Conference Advances in Neural Information Processing systems(NIPS)*, pp. 1324–1332, Vancouver, Canada, December 2010.

[29] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localized crowd counting," *BMVC*, vol. 1, no. 2, 3 pages, 2012.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[31] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," 2017, https://arxiv.org/abs/1612.00220.

[32] Z. Zou, X. Su, X. Qu, and P. Zhou, "DA-net: learning the fine-grained density distribution with deformation aggregation network," *IEEE Access*, vol. 6, pp. 60745–60756, 2018.