Journal of Advanced Transportation

Machine Learning Applications in Transportation Engineering

Lead Guest Editor: Petr Dolezel
Guest Editors: Milos Milenković, Ladislav Routil, and Michael Bazant

Machine Learning Applications in Transportation Engineering

Journal of Advanced Transportation

Machine Learning Applications in Transportation Engineering

Lead Guest Editor: Petr Dolezel
Guest Editors: Milos Milenković, Ladislav Routil, and Michael Bazant

Contents

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 6614242, 10 pages
https://doi.org/10.1155/2022/6614242

Research Article

Demonstration of Smart Railway Level Crossing Design and Validation Using Data from Metro Rail, South Africa

D.C. Tshaai,[1] A.K. Mishra,[1] and Jan. Pidanic[2]

[1]*Department of Electrical and Computer Engineering, University of Cape Town, Cape Town 7700, South Africa*
[2]*Department of Electrical Engineering and Informatics, University of Pardubice, Pardubice 53012, Czech Republic*

Correspondence should be addressed to Jan. Pidanic; jan.pidanic@upce.cz

Received 8 October 2020; Revised 10 December 2020; Accepted 18 January 2022; Published 25 February 2022

Academic Editor: Alejandro Tirachini

Long waiting time at railway level crossings poses a risk on the safety and affects capacity of rail and road traffic. However, in most cases, the long closing time can be prevented by reducing the time lost at a railway level crossing. The emphasis of this study is to present a numerical optimisation algorithm to reduce the time lost per train trip at a railway level crossing. Thus, attributes with the highest impact on the railway level crossing closing time were extracted from the data analysis of rail-road level crossings on the southern corridor of the Western Cape metro rail. Powell's optimisation algorithm was formulated on the minimisation of the time lost at the railway level crossing per trip. Thus, time lost is constrained by the technical and train's traction constraints. The upper and lower bounds of Powell's algorithm were defined by the threshold closing time in addition to the actual and expected probability density functions. The algorithm was implemented in Matlab. Furthermore, the algorithm was trained on 8000 data sets and tested on 2000 data sets. The developed algorithm proved to be effective and robust in comparison to the current state of railway level crossings under study. Thus, the algorithm was validated to reduce the time lost at the railway level crossing by at least 50%.

1. Introduction

A railway level crossing marks a point of shared responsibility between rail and road transport. Both modes of transports have different operational characteristics, but first preference is given to the railway because of its operational complexities. There is a growing concern of unsafe human behaviour at the railway level crossings due to the long closing time of gates on the passage of the train [1]. In addition, the presence of railway level crossings increases the train's travel time in the network, thus constraining traffic capacity. This is often attributed to the imposed speed restrictions as well as train's scheduled stops. Moreover, long waiting time is inevitable in the case of increased railway traffic heterogeneity and volume. Furthermore, a train travelling at the speed significantly lower than the maximum permissible speed results in activation being triggered at the same point as when the train is travelling at the maximum permissible speed. Therefore, a train with slower speed increases the closure duration [2, 3]. Thus, triggering of the railway level crossing from far tends to result in the slower train spending more time over the entire crossing section. As a result, long waiting time poses safety and capacity risks at the railway level crossings [1].

Railway level crossings are both safety critical and sociotechnical. Therefore, the problem with such a system tends to be cross dimensional with a large number of underlying uncertainties. Thus, machine learning has demonstrated superior performance in transit problems and is drawing significant attention in data analysis and optimisation of the railway level crossing parameters [4]. Extensive literature exists in the application of machine learning in the safety and capacity of the railway level crossings in area of safety improvement. For instance, a data consolidation model was developed to determine which railway level crossings can be closed in the United States as part of the effective safety program [5]. This involved the application of an eXtreme Gradient Boosting (XGBoost) algorithm in

determining the railway level crossing closing decision based on the 14 features derived from safety, engineering, economic, environmental, and social aspects. The model was able to achieve an expert judgement with an accuracy of 0.991. On the other hand, an Analytic Hierarchy Process based on the compromise ranking method was applied in railway route planning and design [6]. The results demonstrated that the explicit complex decision-making process can be achieved in railway applications using multicriteria optimisation methods.

Binary programming was applied in the prioritisation of the railway level crossing project, based on the average probability of having m crashes from the n railway level crossings within the corridor [7]. Thus, the average probability obtained from the binary programming was able to select the optimal set of safety improvement actions, which could yield the maximum railway level crossing reliability and mobility. Conversely, the Random Forest algorithm proved effective in the ranking and analysis of the railway level crossing attributes in the selection of the suitable protection type [8]. The algorithm was able to identify the key safety factors with highest influence on the safety improvement and collision prediction at the railway level crossings in Canada. In addition, a full Bayesian analysis has been applied in the selection of the best estimate of the accident modification factors, based on the combination of the likelihood and prior knowledge of countermeasures [9]. The framework was able to discern the anticipated safety benefit of countermeasures in the face of uncertainty across the accident modification factor credible interval.

However, Liang et al. proposed a similar framework on the railway level crossing risk assessment. A probabilistic risk assessment and improved decision was achieved based on the application of Bayesian framework on datasets collected from several railway level crossings in France [10]. Thus, the model indicated that about 81% of the rail and road vehicle accidents resulted in zero fatalities, whilst 19% of accidents were likely to result in fatalities [10]. A combination of Local Estimated Scatterplot Smoothing (LOESS) and Generalized Additive Model (GAM) performed effectively in evaluating the in-vehicle railway level crossing warning system based on a taxi's speed, acceleration, and jerk data [11]. The model revealed that taxi drivers showed improved behaviour with an in-vehicle warning system. Yet speed, acceleration, and jerk difference per multiple transit indicated that the model showed a lack of empirical generalizations of the taxi drivers who used the service.

Machine learning has also been applied in traffic control at the railway level crossings. A multiagent system was proposed in modelling urban traffic control to improve capacity at the railway level crossings [12]. An intersection agent (ISA) was used for the railway level crossing since it marks the intersection of the rail and road. Therefore, fusion of the intelligent ISA control cooperation and genetic reinforcement learning algorithm was proposed to ensure effective cooperation among the rail and road agents. Thus, the ISA control strategy would modify the signal cycle of every intersection with the help of the reinforcement learning in actualizing the local traffic optimisation. The study reported that the intelligent cooperation control is most effective for higher traffic volume; however reinforcement learning allowed the evaluating index to increase when the traffic flow increases and surpasses saturation [12].

An object detection railway level crossing protection system is another field that has seen an increasing application of machine learning. Hence, object parametrisation and localisation usually involve the application of classification learning algorithms and Kalman filtering for object tracking [13, 14]. However, the challenge with these algorithms is evident in video processing because they cannot detect every pixel of the object [13]. Furthermore, machine learning application confirmed that the three best measures for managing railway level crossing accidents can be achieved by the in-vehicle warning system, obstacle detection, and constant warning time [9, 11]. However, constant warning time without optimal railway level crossing closing tends to perpetuate unsafe human behaviour [1, 15]. The impact of long railway level crossing closing time on the railway planning and operation is often significant and, if not monitored, may result in adverse consequences [3, 16]. Moreover, the inability to predict and reinforce minimal railway level crossing closing time is likely to disintegrate the traffic management system [17].

Nikolajevs et al. proposed prediction of the railway level crossing closure duration, based on the train speed measurement by either additional sensors or evaluation of the track circuit's impedance [2]. However, prediction of the closure time does not reinforce adherence to minimal railway level crossing closing time. Alternatively, Work et al. proposed the use of a support vector machine in the prediction of the train arrival time at the railway level crossing. Comparative analysis of the various machine learning algorithms indicated that the Random Forest algorithm provides the best estimation within an appropriate timeframe compared to linear regression and neural network [18]. Likewise, the multitask deep neural network has proven effective in estimating short-term transit delays [4]. Nogushi et al. proposed the reduction of the railway level crossing closing time using an optimal rail-road schedule. The optimal schedule was calculated from a genetic algorithm using time delay for each train at each station as a gene value and the closing time as the fitness value. Thus, the study confirmed the reduction of the railway level crossing blocking time with the changing combinations of the departure time [19]. Moreover, the study concluded that the rail-road waiting time can be reduced through the application of the genetic algorithm on the calculation of the schedule, taking into account the train's location and speed [20]. Thus far, there is limited literature on the application of machine learning or data-centric methods in the optimisation of the railway level crossing closing time.

The present study proposes reducing the time lost at railway level crossing per train trip. Thus, this study contributes to the existing literature in the application of machine learning in the railway level crossing safety and capacity analysis. Moreover, the study demonstrates the significance of data-centric models on rail-road level crossing operations. In this study, railway level crossings

found the southern corridors of the Western Cape metro rail are used to formulate the optimisation of the railway level crossing closing time. Thus, the criterion used by Powell's method [21] is based on the minimisation of the area bounded by the threshold closing time as well as the actual and expected density functions. The proposed method is compared to the current status quo of the railway level crossings in the Western Cape, South Africa. The method showed satisfactory results thus proving its effectiveness.

The structure of the paper takes the following form. Section 2 discusses preliminaries of the railway level crossing closing time and the application of numerical optimisation. The adopted methodological process is outlined in Section 3. Lastly, results and discussions are presented in Sections 4 and 5, respectively.

2. Preliminaries

The railway level crossing depends on the activation and deactivation points denoted by A and C in Figure 1. A simple illustration of the train trajectory at the railway level crossing is shown in Figure 1; thus a track section is delimited by at least two detection points (A or B or C). The detection of the first train axle over A triggers the railway level crossing protection. Similarly, detection of the last train axle over C will withdraw the railway level crossing protection. Thus, railway level crossing closing time is the time difference between the insertion and withdrawal of the protection. The train is expected to brake on the approach of the railway level crossing's activation track (t_a). Furthermore, the train enters into coasting driving mode along the railway level crossing inner area bounded by S0 and S0_1. Lastly, the train accelerates on the exit of the deactivation point (C).

Assume that no vehicles or passengers get trapped inside the railway level crossing once the protection is enabled. The objective variable t (level crossing closing time) is the sum of the contribution of the feature vector x at the ith location on the railway level crossing. The objective function is constrained by the technical and traction parameters shown in Figures 1 and 2, respectively. Hence, optimisation of the railway level crossing time can be expressed as follows:

$$\begin{aligned} &\text{minimise} \sum_{i,k} t_i(x_k) \text{ for } i \in \mathbb{R}_+, k = 1, \ldots, 3, \\ &\quad x_k \varepsilon X_k \text{ (technical constraints stated in Table 1)} \\ &\text{s.t.} \sum_i F_i \geq p,\ p \in \mathbb{R} \text{ (traction characteristics stated in Table 2).} \end{aligned} \tag{1}$$

The term x_k denotes attributes extracted from data collected from the railway level crossings under study. Thus, technical constraints include dwell time, railway level crossing speed restriction, and time delay on the protecting signals. In contrast, traction characteristics include the longitudinal forces and velocity at a location on the railway level crossing. The resultant traction force shown in equation (2) illustrates the relationship between the force acting on the train and the velocity.

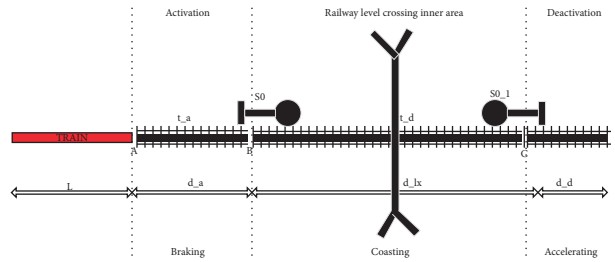


Figure 1: The railway level crossing layout.

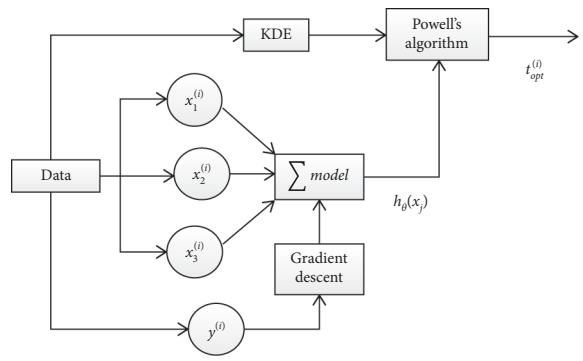


Figure 2: Block diagram of the railway level crossing optimisation solution.

$$\sum F = F_t(v) - R(v, r, \beta) - F_b(v) \tag{2}$$

where

M_J is the train's inertial mass,

$F_t(v) = \mu_t M_j \, \mathrm{d}v/\mathrm{d}t$ is the train's traction force,

$R(v, r, \beta) = c_0 + c_v v + c_a v^2 + M_j(D/1000r) + M_j g \cos\beta$ is the sum of resistive forces,

$F_b(v) = (1/2) M_j \mathrm{d}v/\mathrm{d}t$ is the train's braking force,

(v, r, β) is the train's speed, rail curvature radius, and angle of inclination.

The rail curvature (r) and angle of inclination (β) are different for each railway level crossing. Moreover, coefficients c_0, c_v, and c_a represent the axle-rail friction, mechanical resistance of shaft rotation, and aerodynamic resistance; μ_t is the tractive resistance. Lastly, coefficient D depends on the rail characteristics.

3. Methodology

Railway level crossing closing time is influenced by many attributes, some generic or specific to each system. Generic parameters are always accounted in the design whilst specific parameters receive no attention [3, 22]. However, it is imperative to assess specific attributes to achieve optimal railway level crossing closing time. Thus, specific attributes of the railway level crossings found on the southern corridor of the Western Cape metro rail were analysed. Attributes with highest influence in the railway level crossing closing time were extracted from the analysis. It was found that dwell time, train speed, and time delay imposed on the

protecting signals were the most important features influencing the closing time on the southern corridor [15]. Furthermore, the analysis revealed that the coexistence of at least two of these attributes has a severe effect on the level crossing system's capacity and safety [15].

The diagram in Figure 2 illustrates the method adopted, followed by the detailed overview in Sections 3.1, 3.2, and 3.3. Data of events concerning the operation of the railway level crossings on the southern corridor of the Western Cape metro rail were collected. A total of 10000 separate events of the track occupancy and signal routing involved in the railway level crossing operation were recorded. Only, 'occupied' and 'clear' track occupancy statuses are considered. Thus, track occupation time marks the time at which the first axle of the train is detected on the track section, whilst the track's vacancy or "clear" time marks the time at which the last axle of the train is counted out of the track section.

Railway level crossing closing time is approximated by the time difference between the occupation time of the activation track and clearing time of the deactivation track. In addition, speed of train as it enters the activation track is estimated from the distance, occupation, and release time of the section. Some of the considered railway level crossings have, at most, 15 s time delay on the protecting signals due to lack of braking distance. Thus, total time delay incurred due to the timer on these signals is evaluated from the time the activation track is occupied to the time at which the train exits the track in response to elapsed signal timer.

The regression model is derived from the data, as shown in Figure 2. The data is passed into the kernel density estimation (KDE) algorithm which is used to determine the probability density function of the time spent per train trip. Numerical optimisation (Powell's method) is formulated based on the results of the regression model and KDE algorithm. At last, numerical optimisation is applied to reduce the time lost per train trip as a result yielding optimal railway level crossing closing time.

3.1. Data Analysis. Regression techniques are used to derive the model of the attributes with the highest influence on the railway level crossing closing time. Hence, regression model of railway level crossing closing time y and feature vector x of $n = 3$ features and $m = 8000$ training sets were built in Matlab. Moreover, the model was tested using 2000 data sets. The feature vector represents dwell time, train entry speed, and time delay on the protecting signals denoted by x_1, x_2, and x_3, respectively. Dwell time exhibits a linear pattern whereas others exhibit nonlinear patterns, as shown in Figures 3, 4, and 5. Thus, feature transformation was applied to the nonlinear features.

The hypothesis $(h_\theta(x))$ of the regression model is given in (2) with $q_1(x, \theta)$ and $q_2(x, \theta)$ being the parameterised nonlinear function for train entry speed and time delay on the protecting signal, respectively. The quadratic and exponential feature transformation functions were selected for the train entry speed and time delay. The feature

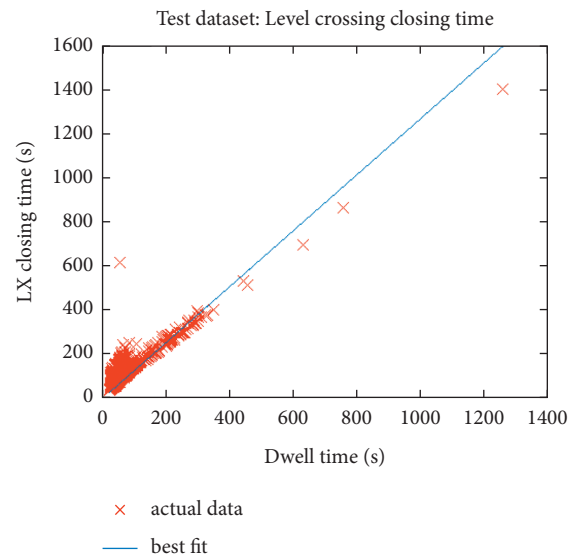


Figure 3: The best fit of the dwell time against railway level crossing closing time.

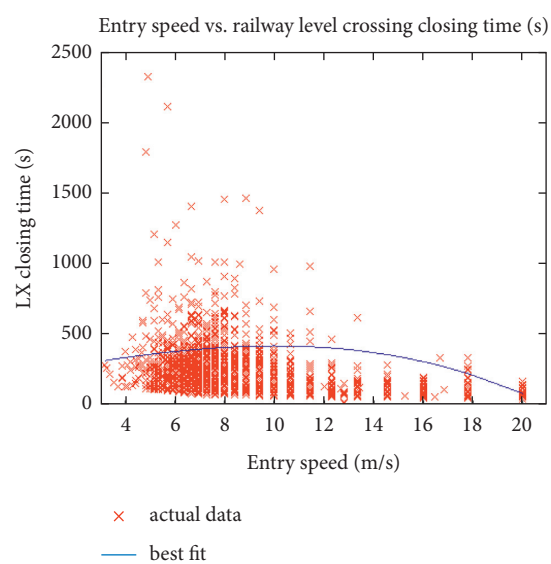


Figure 4: The best fit for the train's entry speed against railway level crossing closing time.

transformation has been introduced to allow for application of the gradient descent and least squares. Thus, expressions of the feature transformation function are shown in (3) and (4). Gradient descent is applied to tune the parameters of the model by minimising the regression cost function over θ as given in (5). In addition, Jacobian leverage is applied in estimating the parameters of the nonlinear feature transformation function.

$$h_\theta(x) = \sum_{i=1}^{3} h_{\theta^{(i)}}(x_i) \tag{3}$$

where $\theta^{(1)} = [\theta_0, \theta_1]^T$, $\theta^{(2)} = [\theta_2, \theta_3, \theta_6]^T$, and $\theta^{(3)} = [\theta_4, \theta_5, \theta_7]^T$; $h_{\theta^{(1)}}(x_1) = \theta_0 + \theta_1 x_1$, $h_{\theta^{(2)}}(x_2) = \theta_2 + \theta_3 q_1(x_2, \theta_6)$, $h_{\theta^{(3)}}(x_3) = \theta_4 + \theta_5 q_2(x_3, \theta_7)$.

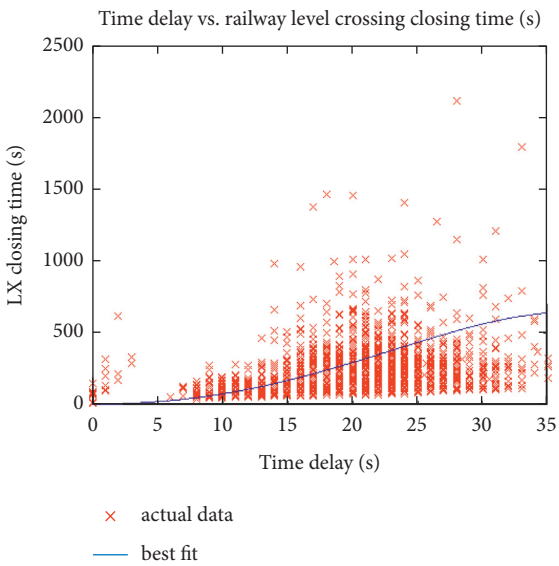


Figure 5: The best fit for time delay against the railway level crossing closing time.

$$q_1(x_2, \theta_6) = \frac{(\max(x_2) + 1 - x_2)}{100}(1 + \theta_6 x_2)^2,$$
$$q_2(x_3, \theta_7) = \exp(\theta_7 x_3), \tag{4}$$
$$\min_{\theta} J(\theta) = \min_{\theta} \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2\right].$$

$\theta = [\theta^{(1)}, \theta^{(2)}, \theta^{(3)}]^T; x^{(i)} = [x_1^{(i)}, x_2^{(i)}, x_3^{(i)}]^T$. Here $h_\theta(x^{(i)})$ represents the estimated closing time based on the derived model and $y^{(i)}$ is the actual closing time.

3.2. Density Estimation. The kernel density estimation method is applied to reconstruct an estimate of the time spent at the railway level crossing per train trip. The reason is that kernel density estimation can converge to the true density faster whilst guaranteeing a smooth output in comparison to its counterparts [23, 24]. Thus, the kernel is a smooth function K which determines the shape of the estimator [21, 25, 26]. In addition, the kernel function partitions dataset of railway level crossing closing time into several bins and estimates the density from the bin count [25, 26]. The Gaussian kernel is chosen for this application and the performance of its estimator $\widehat{p}(x)$ is assessed using risk R and the mean squared error (MSE), expressed in equations (5) and (8), respectively.

$$R = (\text{bias}(\widehat{p}(x)))^2 + \text{var}(\widehat{p}(x)), \tag{5}$$

$$\text{bias}(\widehat{p}(x)) = \frac{1}{2}\delta_k^2 h^2 p'(t) + O\left(h^4\right), \tag{6}$$

$$\text{var}(\widehat{p}(x)) = \frac{1}{nh} p(t)\delta^2 + O\left(\frac{1}{n}\right), \tag{7}$$

$$\text{MSE}(\widehat{p}(x)) = O\left(n^{-(4/5)}\right), \tag{8}$$

where $\widehat{p}(x) = (1/n)\sum_{j=1}^{n}(1/h)K(x - X_j/h)$ is the kernel density estimator, $x = [x_1, x_2, x_3]^T$, $\delta_k^2 = \int t^2 K(t)\mathrm{d}t$, $\delta^2 = \int K^2(t)\mathrm{d}t$, and h is the smoothing bandwidth which depends on sampling size n. In addition, it is assumed that the true density function $p(t)$ is continuous at t and that $h \longrightarrow 0$ and $nh \longrightarrow \infty$. On that note, the estimator $\widehat{p}(t)$ is assumed to convergence in probability to the true density function $p(t)$, i.e., $\widehat{p}(t) \longrightarrow^P p(t)$. The notation var is used for variance.

Similarly, the Gaussian kernel density estimate $\widehat{g}(x)$ of the expected railway level crossing closing time per each train trip is to be populated. The expected closing time is the designed for case, based on the ideal railway level crossing technical parameters and train traction characteristics. Hence, $\widehat{g}(x)$ is populated in accordance with the parameters of the technical and train traction constraints defined in Tables 1 and 2, respectively. Furthermore, threshold closing time is the maximal permissible railway level crossing closing time which takes into account the prevailing constraints. This varies for each railway level crossing. As already mentioned, the recovery of the time lost per passage can improve the railway level crossing safety and capacity. Therefore, time lost by trains at the railway level crossing is represented by the surface area bounded by the threshold railway level crossing's closing time, density function $\widehat{g}(x)$, and density estimator $\widehat{p}(x)$.

3.3. Optimisation Method. Optimisation of the railway level crossing closing time is critical in improving safety and traffic capacity. Thus, optimal closing time is achieved by minimising the time spent by trains on activation, inner area, and deactivation of the railway level crossing. The time spent at the railway level crossings is constrained by the technical parameters listed in Table 1. Dwell time relates to the train's scheduled stop. Hence, the effect due to dwell time is restricted to 10% of anticipated train stop time at the platform. Furthermore, the line speed and railway level crossing speed restriction are applied in accordance with specifications of the southern corridor of the Western Cape metro rail. Lastly, time delay on the protecting signal is limited to an average reaction rate of the driver of 2 s to 4 s.

Additionally, performance constraints are determined by the tractive characteristics on the presiding compartment of the railway level crossing closing time. Thus, resultant force and associated train speed restriction per driving mode are stated in Table 2.

As mentioned above, the time lost at the railway level crossing is represented by the surface area bounded by the threshold railway level crossing's closing time, density function, and density estimator $\widehat{g}(x)$. Then, optimisation of the railway level crossing closing time reduces the denoted minimum surface area. In this case, the time lost at the railway level crossing is obtained by computing numerical or analytical integration, as shown in the following equation:

Table 1: Technical constraints.

Parameter	Abbreviation	Value
Dwell time	t_d	10%
Speed restriction	v_{lx}	30–35 km/h
Time delay	t_i	2s–4s
Line speed	v_L	75 km/h

Table 2: Constraints in each train driving mode.

Driving mode	$\sum F$	v
Braking	$F_t(v) - R(v,r,\beta) - F_b(v) \le 0$	$v \le v_{lx}$
Coasting	$F_t(v) - R(v,r,\beta) - F_b(v) = 0$	$v \le v_{lx}$
Acceleration	$F_t(v) - R(v,r,\beta) - F_b(v) \ge 0$	$v_{lx} \le v \le v_L$

$$I = \int_a^b \int_c^d \int_e^f (\widehat{g}(x) - \widehat{p}(x)) \mathrm{d}x_3 \mathrm{d}x_2 \mathrm{d}x_1, \tag{9}$$

where the integral is evaluated over the 3D bounded region defined by the kernel density estimators $\widehat{g}(x)$ and $\widehat{p}(x)$. The points of interest of the region on which the integral is evaluated are denoted by a, b on the ith plane, c, d on the jth plane, and e, f on the kth plane.

Let $f(x) = \widehat{g}(x) - \widehat{p}(x)$ which can be further composed into $f(x_1, x_2, x_3)$; then numerical integration can be approximated to equation (10) by applying the generalized Gaussian quadrature to evaluate the nodes and weights for the product of the polynomial and logarithmic functions [27]. The generalized Gaussian quadrature formula has been proven to give better results for integration over three-dimensional regions particularly in common applications in science and engineering [27].

$$\begin{aligned} I &= \int_a^b \int_c^d \int_e^f f((x_1, x_2, x_3) \mathrm{d}x_3 \mathrm{d}x_2 \mathrm{d}x_1 \\ &\approx \sum_{l=1}^{L^3} c_l f((x_1)_l, (x_2)_l, (x_3)_l), \end{aligned} \tag{10}$$

where $c_l = w_1^i w_2^j w_3^k (b-a)(d-c)(f-e)$; $(x_1)_l = (b-a)\xi_i + a$; $(x_2)_l = (d-c)\eta_j + c$; $(x_2)_l = (f-e)\zeta_k + e$. The term L is the number of selected point set over which the integration is to be evaluated.

Here, ξ_i, η_j, and ζ_k are the node points, and w_1^i, w_2^j, and w_3^k are the corresponding weights in one dimension. Several assumptions are postulated; features such as dwell time, train entry speed, and time delay on the protecting signal are nonnegative. Furthermore, the constrained set is nonempty and the objective function $f(x_1, x_2, x_3)$ is finite. Thus, numerical optimisation can be formulated as follows:

$$\begin{aligned} &\min_{x_j \in R_+} \sum_{l=1}^{L^2} c_l f((x_1)_l, (x_2)_l, (x_3)_l) \text{ for } j = 1, 2, 3 \\ &\text{subject to} \quad \begin{aligned} &\arg\min\left(h_{\theta^{(1)}}(x_1)\right) \le t_d, \arg\max\left(h_{\theta^{(2)}}(x_2)\right) \le v_{lx}, \\ &\arg\min\left(\left(h_{\theta^{(3)}}(x_3)\right)\right) \le t_i, F_t(x_2) - R(x_2, r, \beta) - F_b(x_2) \le 0. \end{aligned} \end{aligned} \tag{11}$$

Powell's optimisation method is applied to minimise the time lost at the railway level crossing per train trip. The Powell method is a single-shot and fast converging method which attempts to find the local fit-statistics minimum nearest to the starting point [28]. The advantage of using the Powell method is its robust direction set. Hence, it will move in one direction until it finds the minimum [28, 29]. Furthermore, it is a gradient-free minimisation algorithm and therefore it does not require the objective function to be smooth [29–31]. In this application there are 3 design variables x_1, x_2, and x_3; thus the Powell algorithm can converge faster than numerical optimisation algorithms. In addition, it has been proven effective for problems with less than 10 design variables [28]. Although the algorithm can find multiple local optima there is no guarantee that it will find the global optimum [29]. Nonetheless, it can produce better results for this application.

The new unconstrained optimisation algorithm (NEWUOA) software of the Powell method is implemented in Matlab due to its efficacy in minimising a noisy objective function [30]. NEWUOA uses a truncated conjugated gradient algorithm to find the minimum m_k within the trust region [32]. The efficiency of the algorithm is derived from the intermingled trust region iteration and model iteration. Thus, objective of the trust region step is to find the better objective function value whilst the model iteration improves the model [32]. Assume a starting point x_s for each design variable with the objective function $f(x_s)$ and initial interpolation set Y containing x_s. The initialisation stage assigns $k \longleftarrow 0$, Δ_k and $x_k \longleftarrow \arg\min\{f(y_i), y_i \in Y\}$. The initial point of the NEWUOA x_k is chosen as a point with lowest objective function value. The result of optimisation is that the step s_k is added to x_k to give the new point; $x_k^+ \longleftarrow x_k + s_k$. The length of s_k can either become the trust region iteration or model iteration. Thus, if the length of s_k is the too short, it indicates that the model must be improved; therefore it is referred to as model iteration. Otherwise, the length of s_k becomes the trust region iteration.

In the trust region step, where $f(x_k^+) < f(x_k)$, the objective function value is replaced by the new point x_k^+, whereas in the model iteration, a point x_k^+ will not be added to the model. Instead, a new point will be calculated and will replace the point furthest from x_{opt}. Moreover, new point is chosen in such a way that it improves minimisation m_k. The algorithm uses several parameters such as the trust region radius Δ_k, which is related to initial current radius (ρ_{beg}), lower limit of the radius (ρ_{end}), and upper limit of the radius (ρ_k). The normal trust region radius Δ_k is used to limit the step length. In addition, ρ_{beg} is used to keep enough distance between the interpolation points $(> ((2n+1)+n)^2, n = 3)$ in the initial model so that it is still accurate in case of presence of errors in the evaluation of f. The following parameters were used for the NEWUOA model.

4. Results

The results of the study are presented as follows. Section 4.1 entails the results of the data analysis followed by the kernel density estimation and optimisation results in Sections 4.2 and 4.3, respectively. Validation of the proposed method is presented in Section 4.4. Throughout, the study training dataset of 8000 samples and test dataset of 2000 samples have been used.

4.1. Data Analysis. The gradient descent algorithm has been applied to estimate the parameters of the model listed in Tables 3 and 4. Similarly, the learning curve shown in Figure 6 indicates that the regression algorithm converges to the local or global minima. The deviation in the training and test cost functions after convergence is not vast; thus there is no indication of overfitting or underfitting. However, the learning curve does not give an indication of the impact of the residuals. Hence, error analysis in Table 5 indicates that outliers, due to feature transformation, have been heavily penalised. Yet, the model achieves accuracy of 78.2 % which is relatively good given the data.

4.2. Kernel Density Estimation. The bias-variance kernel density estimator trade-off is used to determine the amount of smoothing required. The optimal amount of smoothing minimises the risk and consequently reduces the mean square error. The smoothing bandwidth is chosen to be a multiple of the sampling rate. The iterations of smoothing bandwidths are run to determine the bandwidth that minimises the risk and mean square error by visually observing the estimator output. The summary of results is shown in Table 6. It can be observed that the appropriate smoothing bandwidth for this application is 0.08 since the spurious effect of the data is not masked. Furthermore, density estimators of the features with significant impact on the railway level crossing time per train trip are shown in Figure 7. The culmination of the estimators would result in the 3D output; however for this analysis they are separated. The contribution of the dwell time is less significant on the first lobe in comparison to the entry velocity and time delay.

4.3. NEWUOA Method. The NEWUOA software which is a Powell new unconstrained optimisation algorithm was used to minimise the area of a region between the density estimates of the features with highest influence on the railway level crossing time. The NEWUOA algorithm developed by Powell is run on Fortran and is interfaced to Matlab [29, 30]. The sensitivity analysis of the algorithm is shown in Table 7. Herein, the time consumption and number of iterations required for convergence at chosen initial points $[(x_1, y), (x_2, y), (x_3, y)]$ are also presented.

The sensitivity analysis of the optimisation solution indicates that the a priori information can significantly reduce the computation time and the number of iterations required. Moreover, choosing the initial point in between the far extremes reduces the computation time and number of iterations. The minimum required number of iterations $((2n+1)+n)^2$ has been kept, to increase chances of obtaining the global optima.

4.4. Validation. The efficacy of the solution is validated on the dataset of 2000 samples from the eight railway level crossings under study. The actual and optimal time lost at the railway level crossings are shown in Table 8. Thus, results indicate that the optimisation algorithm can achieve, at most, 40% reduction of the time lost at the railway level crossing. However, the effectiveness of the optimisation algorithm is not the same for each railway level crossing.

5. Discussion

In this study, it has shown that long closing time at the railway level crossings is attributed to three features. The features identified to have highest impact are dwell time, time delay on the protection signals, and train entry speed. The regression model confirmed the relationship of these features and the railway level crossing closing time. The model has shown that dwell time is directly proportional to the railway level crossing closing time. Since it is inevitable to remove the platform at activation of some railway level crossings, the algorithm imposed a delay on the triggering of the protection system, such that the impact is well mitigated. Similarly, train entry speed plays a significant role in the railway level crossing system. The model indicated that the railway level crossing closing time decreases with an increase in train speed following a hyperbolic trend. Train travelling over the railway level crossing at speed lower than the permissible speed results in longer closing time. Lastly, time delay imposed on the protecting signal tends to introduce a delayed train driver response and consequently increases the closing time.

The model has shown that the railway level crossing closing time increases with time delay on the protecting signal in an exponential manner particularly where automatic routing is applied. The presented railway level crossing optimisation incorporated regression models of the identified attributes. In addition, the optimisation algorithm makes use of the density functions of the features to define the time lost at the railway level crossing per train trip. The threshold closing time, as well as the actual and expected closing time density function, defines the lower and upper boundaries of the Powell algorithm. Thus, recovery of the time lost at the railway level crossing is posed as the minimisation of the area bounded by the threshold railway level crossing closing time, actual, and expected density functions.

The optimisation of the railway level crossing closing time is subjected to train speed (at least minimal permissible speed), dwell time (at most 10% of the anticipate dwell time value), and time delay of at least the average driver reaction time. Furthermore, train traction characteristics constrained the objective function. The developed optimisation algorithm was trained on 8000 data samples. The algorithm exhibited high performance for the number of training datasets. However, performance can still be improved by increasing the number of training datasets, as well as introducing additional attributes. Validation of the algorithm proved that the features identified have a significant impact on long level crossing closing time. Moreover, the optimisation algorithm achieved at least 50 % decrease in the time lost at the railway level crossings on 2000 test datasets.

Overall, the algorithm has proven to improve safety and capacity at the railway level crossings by reducing the time lost per train trip. Hence, optimal railway level crossing closing time is feasible if the technical and train tractive constraints are adhered to. However, the inconsistency of

Table 3: NEWUOA parameters.

Description	Term	Value
Initial radius	ρ_{beg}	0.01
Lower limit of the radius	ρ_{end}	0.01
Upper limit of the radius	ρ_k	0.3
Trust region radius	Δ	0.05

Table 4: Parameters of the trained regression model.

	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
Model	0.05	1.27	11.28	0.701	18.06	9.934	−1.23	−0.15

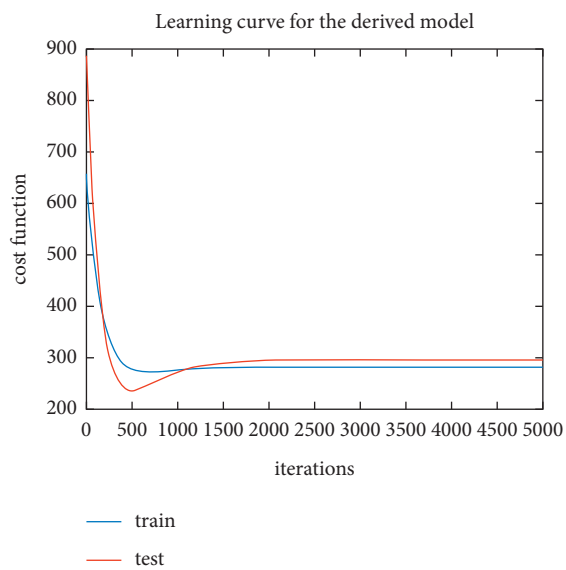


Figure 6: Learning curve for the regression model.

Table 5: Error analysis.

Iterations	Error		Accuracy
	Train error	Test error	
500	105.23	100.77	0.5820
1000	91.08	90.182	0.6530
3000	78.97	69.61	0.714
4000	78.34	68.97	0.734
5000	78.05	68.39	0.782

Table 6: Performance analysis of kernel density estimates.

Smoothing bandwidth	Observation
0.001	Undersmoothed
0.008	Undersmoothed
0.01	Undersmoothed
0.08	Just right
0.1	Oversmoothed
0.8	Oversmoothed
1	Oversmoothed

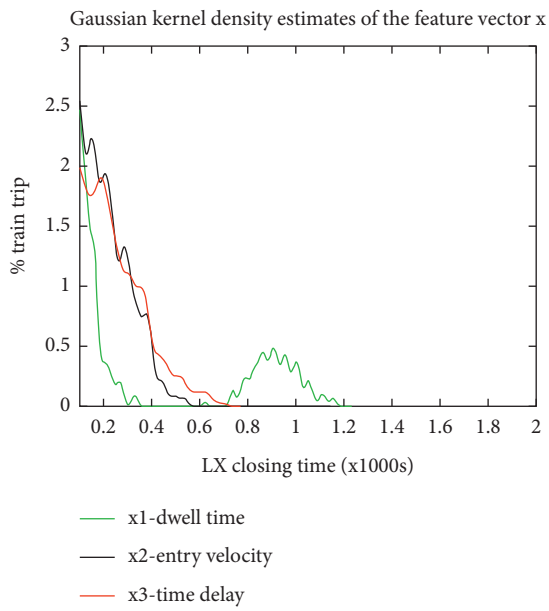


Figure 7: Gaussian kernel density estimator of the features with highest impact on the railway level crossing closing time.

Table 7: Sensitivity analysis of the Powell optimisation using NEWUOA software.

Initial condition	Time (s)	Iterations
(0,30),(8.5,30),(2,30)	13.430	10000
(15,40),(7.1,40),(4,40)	8.005	7060
(30,50),(5,50),(10,50)	6.233	5002
(35,60),(9,60),(14,60)	2.511	1890
(35,70),(4,70),(12,70)	2.084	1055
(30,80),(4.2,80),(10,80)	1.552	1010
(40,90),(5.2,90),(14,90)	2.126	1580
(35,100),(6,100),(10,100)	2.905	2806
(45,120),(4.3,120),(10,120)	3.057	3500

Table 8: Validation of the recovered time lost at the railway level crossing.

Level crossing	Line	Actual	Optimal
Albertyn rd	1	0.3167	0.12192
Albertyn rd	2	0.3678	0.14381
Austell rd	1	0.4156	0.16694
Austell rd	2	0.3167	0.12669
Beach rd	-	0.4001	0.15163
Kalkbay rd	1	0.4224	0.14872
Kalkbay rd	2	0.5774	0.20951
Military rd	1	0.6367	0.23717
Military rd	2	0.5786	0.21839
Uxbridge rd	1	0.2852	0.11460
Uxbridge rd	2	0.6131	0.24483
White rd	1	0.5364	0.20917
White rd	2	0.6123	0.24495
York rd	1	0.4035	0.16183
York rd	2	0.4044	0.16092

the optimiser on respective railway level crossings suggests that there may be unique features other than those considered. Therefore, increasing the number of features may improve the generality of the algorithm. In addition, the solution does not take into account mechanical failures of the train at the railway level crossing.

6. Conclusion

The present study developed an optimisation method to reduce the time lost at the railway level crossing per train trip. As a result, optimal railway level crossing closing time can be achieved. The algorithm used minimisation of the area bounded by the threshold closing time, as well as the actual and expected density function of the railway level crossing closing time per train trip, as the criterion. Powell's optimisation algorithm was trained on 8000 datasets and has shown to converge to the optimal solution. Furthermore, the algorithm was validated. The reduction in the time lost at the rail-road level crossing is at least 50% on 2000 test datasets. The proposed solution is critical in ensuring improved safety and capacity at the railway level crossings. However, the algorithm is still selective in the optimisation of some of the railway level crossing closing times, thus suggesting that improvement can be made to achieve consistent results.

Data Availability

The data are available on request.

Conflicts of Interest

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

Acknowledgments

The work was supported from ERDF/ESF "Cooperation in Applied Research between the University of Pardubice and Companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans)" (no. CZ.02.1.01/0.0/0.0/17_049/0008394). Furthermore, the authors would like to acknowledge Thales Western Cape Projects and PRASA Western Cape for granting access to the data used in the study.

References

[1] A. Kumar, "Human behavioural aspects of level crossing safety with special reference to Indian railway," *Jordan Journal on Mechanical and Industrial Engineering*, vol. 6, no. 1, pp. 37–43, 2012.

[2] A. Nikolajevs and M. Mezitis, "Level crossing time prediction," in *Proceedings of the International Scientific Conference on Power and Electrical Engineering of Riga Technical University*, Riga, Latvia, October 2016.

[3] M. Sojka, "The railway level crossing: synergy effects between rail and road infrastructure capacity," in *Proceedings of the wiss Transport and Research Conference*, no. 16, Riga, Latvia, October 2016.

[4] F. Sun, A. Dubey, C. Samal, H. Baroud, and C. Kulkarni, "Short-term transit decision support using multi-task deep neural networks," in *Proceedings of the IEEE International*

Conference on Smart Computing, pp. 155–162, Taormina, Italy, June 2018.

[5] S. Soleimani, S. R. Mousa, J. Codjoe, and M. Leitner, "A comprehensive railroad-highway grade crossing consolidation model: a machine learning approach," *Accident Analysis & Prevention*, vol. 128, no. 128, pp. 65–77, 2019.

[6] D. Baric, Z. Radacic, and C. Danko, "Implementation of multi-criteria decision-making method in selecting the railway line construction," Edited by M. Zanne, D. Fabjan, and P. Jenček, Eds., in *Proceedings of the ICTS 2006 Transportation Logistics in Science and Practice*, vol. 5, Univerza v Ljubljani, Fakultet za Pomorstvo in Promet, Portorož, Slovenia, 2006.

[7] J. Arellano, A. Mindick-Walling, A. Thomas, and A. Rezvani, "Prioritizing of infrastructure investment for safety projects: a corridor-level approach," *Transportation Research Board*, vol. 96, 2017.

[8] C. Yang, E. Trudel, and Y. Liu, "Machine learning-based methods for analysing grade crossing safety," *Cluster Computing*, vol. 20, no. 2, pp. 1625–1635, 2017.

[9] S. Washington and J. Oh, "Bayesian methodology incorporating expert judgment for ranking countermeasure effectiveness under uncertainty: example applied to at grade railroad crossings in Korea," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 234–247, 2006.

[10] C. Liang and M. Ghazel, "A risk assessment study on accidents at French level crossings using bayesian belief networks," *International Journal of Injury Control and Safety Promotion*, vol. 25, no. 2, pp. 162–172, 2018.

[11] A. Skoufas, S. Basbas, J. Grau, and G. Aifadopoulou, "Analysis of in-vehicle warning system for rail-road level crossings: case study in the city of thessaloniki," *Periodica Polytechnica Transportation Engineering*, vol. 49, pp. 1–18, 2019.

[12] Z. Yang, X. Chen, Y. Tang, and J. Sun, "Intelligent cooperation control of urban traffic networks," *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 1482–1486, 2005.

[13] H. Salmane, L. Khoudour, and Y. Ruichek, "A video analysis based railway-road safety system for detecting hazard situations-methodology at level crossing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 20, pp. 596–609, 2015.

[14] X. Li, S. Shen, and S. Jiancheng, "Object tracking using adaptive kalman filter combined with mean shift," *Optical Engineering*, vol. 49, no. 2, 2010.

[15] D. Tshaai, *Optimisation of the Rail-Road Level Crossing Closing Time in a Heterogeneous Railway Traffic: Towards Safety Improvement (South African Case Study)*, pp. 30–60, University of Cape Town, Cape Town, South Africa, 2020.

[16] J. Tornquist, "Computer-based decision support for railway traffic scheduling and dispatching: a review of models and algorithms," *ATMOS Workshop on Algorithmic Methods and Models for Optimization of Railways*, vol. 1, no. 5, 2005.

[17] P. Murali, M. Dessouky, F. Ordóñez, and K. Palmer, "A delay estimation technique for single and double-track railroads," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 4, pp. 483–495, 2010.

[18] D. Work, W. Barbour, and R. Wang, *Improving Railroad Grade Crossing Safety: Accident Prediction of Train Arrival Times for Emergency Response Management and Driver Decision Support*, pp. 1–69, Roadway Safety Institute, Alexandria, Virginia 22312, USA, 2019.

[19] Y. Nogushi, H. Mochizuki, S. Takahashi, H. Nakamura, S. Kaneko, and M. Sakai, "Blocking time reduction for level crossings using genetic algorithm," *Computers in Railway*, vol. 88, no. 10, pp. 299–308, 2006.

[20] I. Alps, M. Gorobetz, and A. Levchenkov, "Algorithm for increasing traffic capacity of level crossing using scheduling theory and intelligent embedded devices," *Riga Technical University Electrical, Control and Communication Engineering*, vol. 29, no. 1, pp. 129–136, 2013.

[21] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, pp. 38–40, Chapman & Hall, London, New York, 1986.

[22] P. Meyer, R. Chavagnat, and F. Bourgeteau, *Computation of the Safe Emergency Braking Deceleration for Trains Operated by ETCS/ERTMS Using the Monte Carlo Statistical Approach*, World Congress on Railway Research, Birmingham, UK, 2011.

[23] L. A. Wasserman, *All of Statistics: A Concise Course in Statistical Interference*, pp. 314–316, Springer, New York, 2004.

[24] L. A. Wasserman, *All of Nonparametric Statistics*, pp. 133-134, Springer-Verlag, Germany, Berlin, 2006.

[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, pp. 208–218, Springer, Germany, Berlin, 2017.

[26] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Annals of Statistics*, vol. 38, no. 5, pp. 2916–2957, 2010.

[27] S. Jayan and K. V. Nagaraja, "A general and effective numerical integration method to evaluate triple integrals using generalized Gaussian quadrature," *Procedia Engineering*, vol. 127, pp. 1041–1047, 2015.

[28] M. J. D. Powell, "Developments of NEWUOA for minimization without derivatives," *IMA Journal of Numerical Analysis*, vol. 28, no. 4, pp. 649–664, 2008.

[29] M. A. Belen, N. Echebest, and E. A. Pilotta, "Active-set strategy in Powell's method for optimization without derivatives," *Computational and Applied Mathematics*, vol. 30, no. 1, pp. 171–196, 2011.

[30] M. J. D. Powell, "The NEWUOA Software for unconstrained optimization minimising without derivatives," *Nonconvex Optimization and Its Applications*, Springer, vol. 83, , 2006.

[31] J. Nocedal and S. J. Wright, *Numerical Optimisation*, Springer, no. 2, , pp. 270–302, Berlin, Germany, 2006.

[32] P. Olsson, *Methods for Network Optimization and Parallel Derivative-free Optimization*, pp. 176–180, Linkoping University, Linköping, Sweden, 2014.

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 8858756, 7 pages
https://doi.org/10.1155/2022/8858756

Research Article

The Train Delay Model Developed by the Genetic Programming Algorithm

Tomas Brandejsky

Department of Software Technologie, Faculty of Electrical Engineering and Informatics University of Pardubice, Pardubice 532 10, Czech Republic

Correspondence should be addressed to Tomas Brandejsky; tomas.brandejsky@upce.cz

Received 25 September 2020; Revised 30 November 2020; Accepted 17 December 2021; Published 31 January 2022

Academic Editor: Alessandro Severino

The paper discusses the problem of probability distribution category identification of train delay data by a genetic programming algorithm. This train delay frequency function and the probability distribution simply derived from it are significant to train traffic modelling and management. The genetic programming algorithm was used as an uninformed tool to prevent the influence of a priori information, which should be biased. The real traffic data were aggregated into predefined bins and then the frequencies of the individual delays were computed. The genetic programming algorithm was used in the next step as a symbolic regression tool to discover their frequency function in the form of an algebraic expression. The results concluded that although data has no known distribution, their distributions are similar to exponential ones.

1. Introduction

Train delays represent inconvenience of the rail transport. They are unwelcome not only in passenger rail transport but also in freight rail transport where delays disrupt logistic chains. The most sensitive is the combination of passenger and freight rail operations for a large difference in the passenger and freight train dynamics which results in complicated train operation scheduling and management.

1.1. Relevant Research. Rail transport delay modelling is significant for identification of delay causes which is essential for this delay reduction or even elimination. The delay modelling is also necessary for rail traffic modelling, control, scheduling, and future rail network state prediction. It is also necessary to mention that delays are typically computed for trains, not for passengers [1]. It is significant if we reason that the preservation of the train change possibility spreads a delay to other trains and can increase the magnitude of this delay, e.g., for passengers transferring between lines. A similar situation occurs in freight train transport if the wagons are switched between different trains.

The train delay models gain different forms. They can be based on a precise model of the rail network, train dynamics, and train timetable [2] on one side and a simplified probabilistic model on the other. These simplified probabilistic models are frequently applied for their easier evaluation and maintenance. For their application, it is required to identify probability distribution functions for all variables of the model. These distribution functions or frequency functions can be obtained from the model or from the data measured in real rail traffic. Analysis of the real traffic data represents a task of large or even Big data size problem. There are many variables influencing railway related processes and a huge number of records. To solve the traffic delay problem by probabilistic techniques, it is necessary to compute train delay distribution or frequency functions applicable in train delay management and train delay models.

It is impossible to reason about the development of the "absolutely" precise model for lack of related data and their uncertainty caused, e.g., by measurement errors. For herein

presented research, only the data about delayed trains were available, but not the data about nondelayed trains, their load, weather, large sport or cultural activities, and so on, and especially no data about operational dependencies like prescribed waiting rules between passenger trains in each station. Only the common preferences between different train categories were defined.

The main reason of this research was to find delay frequency function (FF) for delayed trains of a given category. Such a function can be simply transformed into relative frequency and probability density functions. The infrastructure owner was using exponential distribution in its models, but there was suspicion of imprecision of this model and the possibility that different train categories should be described by different distribution models due to different dynamics and operation rules. The genetic programming algorithm was chosen to find FFs describing observed data as objective tools with little a priori information.

The use of genetic programming algorithm (GPA) was decided on the basis of the observation in [3] that no standard probability distribution category FF describes data perfectly and the closest is an exponential distribution.

1.2. Genetic Programming Algorithm Evolution. Genetic programming (GP) was developed by J. Koza [4] on the basis of genetic algorithm (GA). Koza continued research of John Holland on "Adaptation in Natural and Artificial Systems" with the idea that programming can be transformed into optimization of a randomly generated population of individuals representing instructions and parameters. The state space of GPA is discrete in view of the change of operators (called functions) like addition, subtraction, or substitution during mutation and crossover operations. Latter many alternatives to GPA were developed like grammar evolution (GE), gene expression programming (GEP), or Cartesian GP (CGP). GE tries to reach better efficiency by application of a user-specified grammar (usually a grammar in Backus–Naur form) [5]. Gene expression programming (GEP) represents a different approach suitable to solving the symbolic regression (SR) problems [6], applying not only replication and mutation operations but also the transposition and recombination ones. A graph-based Cartesian GP (CGP) [7] is a GP algorithm where the candidate solutions are represented as a string of integers of a fixed length that is mapped to a directed graph. CGP can efficiently represent such structures as mathematical equations, computer programs, or neural networks. Another modification of genetic programming is represented by a hybrid single node genetic programming (SNGP) [8] [9]. SNGP is a rather new graph-based GP system that evolves a population of individuals, each consisting of a single program node. Similarly to CGP, the evolution is provided by a hill-climbing algorithm using a single reversible mutation operator. The SNGP represents a very promising development of GPs. Also, the original GPAs were widely studied since its introduction, for example, in ([10–12]), and these works have allowed its expansion. A symbolic regression takes specific position between the GPA applications. The goal of SR is to find a function modelling training data. As a technique of uninformed computer learning, the GP has large application potential.

SR is also easily verifiable, and thus it serves as a test bed frequently. Unfortunately, till now, symbolic regression has had some weaknesses, especially problems with large training data sets, e.g., in Big data applications and low efficiency if human comparative quality results are required. Possible ways of their elimination will be discussed further in this study. According to the paper [13], genetic programming (GP) is a particularly interesting machine learning (ML) algorithm when dealing with symbolic regression.

The GPAs directly evolve mathematical expressions [14], typically represented by a tree structure. While GP is understood to be capable of generating white-box models, i.e., human-interpretable expressions in simple form, the evolved models are often overcomplicated and far from being interpretable [15]. These problems tend to improve, especially in the areas of accurate symbolic regression, hybrid evolutionary algorithms, and some other approaches. Depending on the purpose, any rapidly developed or precise model is searched. While linear regression means findings of the coefficients of the chosen function to best fit the data, symbolic regression searches for a suitable function. If the linear regression result is not good, it is possible to choose a different function. The quality of the regression result depends on the selection of this function. Symbolic regression is capable of going far. Its goal is to find such a function to fit the data, not only its parameters. Since the first application of genetic programming [4], many studies and approaches to solve this problem have been published.

Except the abovementioned problems, many authors published in their works the need to use large populations of thousands or more individuals. This is caused by the need to optimize many constants and the sensitivity of individual selection on them. If the modelled problem is complicated and described by training data containing tens or even hundreds of variables (e.g., Big data problems), it is necessary to reduce this amount to prevent loss of GPA efficiency [16] to keep reasonable processing time. In these cases, the efficiency of machine learning algorithms including GPA significantly decreases and it is better to use a separate algorithm to identify significant features, e.g., by the feature selection algorithm [17].

1.3. Hybrid Evolutionary Algorithms. Unpublished work [11] prefigured small but significant research in an area of hybrid evolutionary systems combining GPA with other techniques with the intent to eliminate some of their weaknesses. This work is continued in the paper [18]. Unfortunately, the main research areas of both authors of this work moved out of evolutionary programming and thus these papers were not extended by them. Luckily, they found a continuation in [19]. Raidl assigned multiplication parameters to each node of the parse tree and optimized their magnitudes by the method of least squares. The nonlinear optimization method was used in [18, 20]. The problem of these early works was in large computational requirements

which have allow only a few steps of the time-consuming nonlinear optimization to be applied to each new solution to keep the total running times acceptable. Particle swarm optimization (PSO) was used for arbitrary constant magnitudes optimization in [21]. PSO is a population using a computational optimization method developed by Eberhart and Kennedy in 1995 [22], inspired by the social behaviour of bird flocking. The system is initialized with a population of random solutions and searches space by the coordinated movement of a particle swarm. Accurate SR tries to eliminate well-known problems of original early GPA applications. They were less efficient, overcomplicated, and imprecise. The accurate SR as it was described in [23] applies structural risk minimization based on the Vapnik–Chervonenkis dimension to estimate the difference between the generalization and the empirical error. The problem of these algorithms is that they move GPAs close to overfitting limitation known from the other machine learning algorithms. These works prove that in the hybrid evolutionary algorithm it is possible to apply in the hybrid algorithm any optimization tool in combination with GPA. Thus, there evolutionary strategy of genetic algorithms can be placed too, as in this work.

2. Materials and Methods

2.1. Analysed Train Delay Data Set. To analyse train delays on the Czech national railway network, the data describing train delays within three months of 2011 was used. Originally, the data were stored in Oracle data warehouse. These data were analysed in [3] with recommendation to use exponential distribution in models.

There were problems with trains free of delays because such trains were not presented in the filtered data provided for the research. Thus, the number of observations of the train passing without delay was smaller than in reality, and they were not used to identify delay FF.

Regardless of the above-described limitations, the amount of processed data was not totally satisfying the attributes of Big data, but it was difficult for processing. Big data are characterised by so called 3 or 4 v's. They are mean volume, variety, velocity, and veracity. The meaning of these terms is described as follows:

Volume—The quantity of generated and stored data. The Big data applications work with large amount of data that are hard to process by standard technology.

Variety—Variety means the type and nature of the data. The data processed by the Big data systems are semi-structured or unstructured. It makes it difficult to process them by standard strongly typed relational database management system (RDBMS).

Velocity—Many Big data applications require high speed reactions and fast processing because they operate in real time. There is also the problem of the continuous incoming of a new data.

Veracity—Referring to data volume and their value, the data can vary too fast for accurate analysis.

Because only a small, time-limited sample of data (3 months) was analysed, the volume of analysed data was relatively small from Big data viewpoint, but its size was still many gigabytes. The data were also well structured. The velocity in this off-line analysis is not significant. Nevertheless, with respect to the future possibility of full data collection analysis, the Big data technology was used. There it is possible to imagine online analytics (with high requirements on velocity) and processing of full data collection representing all train movements in the past 20 years.

For the herein presented analysis, the basic programming model of Big data processing called MapReduce was used (selection of data related to the analysed category and aggregation on the basis of delay magnitude into the relevant bin).

There it is justifiable to expect distinct results for different train categories (Table 1) due to different operational rules like maximal speed limit, typical weight and length of the train limiting maximal acceleration, waiting for connections (and concluding delay propagation) in the case of passenger trains, and priority in access to railway line. Especially, the last two influences strongly determine delays. Figure 1 demonstrates an example of input data for passenger train categories.

2.2. Data Preprocessing. The original data set was transformed from the Oracle database into comma-separated values (CSVs) dale file and imported into the Apache Spark application. This application was written in *Python* using Jupyter Notebook and served to select data related to the studied train type and reduce data volume by their transformation into counts of train passes across measurement points with a specified delay. The following step was application of symbolic regression to model FF describing the data.

2.3. Used GPA-ES Evolutionary Algorithm. In this research, a modification of the hybrid GPA-ES algorithm for algebraic dependency modelling is used (standard GPA-ES was designed for symbolic regression of ordinary, linear, or nonlinear differential equations—e.g., 10, 24, or 25). The algorithm evolves equations on the basis of the minimization of residual errors—fitness. This algorithm combines the standard GP algorithm for solution structure development and an evolutionary strategy (ES) algorithm for optimization of parameters of each individual in a GPA population. Such a design of the hybrid evolutionary algorithm prevents situations when a well-structured solution (e.g., well-composed equation) but with wrongly estimated constants (coefficients) is eliminated from the population and replaced by an individual of worse structure but better fitted constants, which has worse evolutionary potential. A comprehensive introduction to GPA can be found in [26]. A complex review of the hybrid evolutionary algorithms is in [27], and an explanation of symmetric one-point crossover is described in [28]. Two different variants of crossover increase the efficiency of the Algorithm 1.

The size of the GPA population and the size of the ES populations related to each individual related to the GPA population are the most significant parameters of the GPA-ES algorithm. The influence of the population size was studied in many publications, for example the study conducted in [2].

Table 1: Explanation of the analysed train category abbreviations and the processed amount of data.

Abbreviation	No. of records in the analysed database	Train category
EC	430228	EuroCity express
Ex	374674	Express train
IC	211926	InterCity express
Lv	1118057	Locomotive train
Nex	612071	Freight fast express
Mn	377838	Service train
Os	8220824	Passenger train
Pn	1271189	Unit freight train
R	2454901	Regional train
Sp	550954	Commuter train

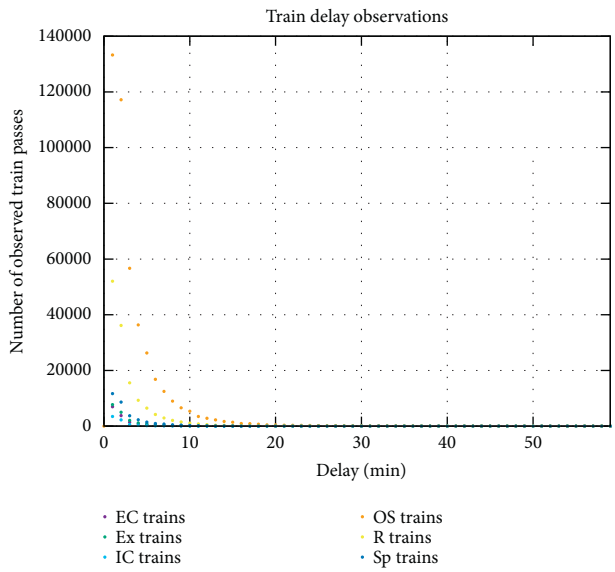


Figure 1: Example of train delay data histograms distinguished by train category.

Small GPA populations forces evolutionary pressure, and in the case of specific conditions it might speed up evolution. On one hand, in small populations there is an increased risk of getting stuck in local optima instead of global ones. Very large populations have problems with low speed and low efficiency of evolution frequently. On the other hand, they bring less dispersion of needed evolutionary cycles and higher reliability of solution discovery. Because extremely small GPA populations bring big dispersion of needed evolutionary cycles, it is difficult to predict the needed computational time. In the case of ES populations, analogical reasons are valid only if there are eliminated random influences of task switching and other sources.

During the initial stage of processing, the records about the delays of a selected type of train were reduced into a vector representing the delay discrete distribution FF by eliminating temporal and spatial information. There are present multiplication, exponential, divide, and factorial functions in the set of GPA functions to allow "discovery" of exponential or Poisson distribution related frequency functions (and similar ones), the candidates to possible solution. The number of reasoned data categories was limited to 20. In the case of large training data vectors, the less significant (outlier) data representing large delays were not reasoned, e.g., in the case of 1-minute delay resolution. Thus, the training data vector was a vector of 20 pairs (delay time and frequency).

Larger delay bins, like the extreme 100 minute one, contain less amount of accumulated noise, but such large bins also lose information. They lose information about delay distribution, and thus they might cause wrong results.

The use of grouping many delays, e.g., from interval <1, 30> minutes, can decrease noise caused especially by small number of samples in the given bin. On the other hand, such grouping represents a significant loss of information. It tends to be a simple function to find and in the extreme case, the linear FF instead of the exponential one can be found.

Because Figure 1 points to functions similar to exponential or Poisson distribution ones, the function set for GPA contains functions that allows to reconstruct them and build a class of similar ones. Thus, there has been a present factorial, but it has a limited scale of argument magnitudes due to the limitations of number representation in computer. Also, the number of distinguished the most significant (the smallest) delays from the interval <1, 20> minutes was reasoned. Exponential or power functions, divide or inverse functions ($1/x$), multiplication, addition, and subtraction are useful in FF regression, and thus they were incorporated

```
(1) FOR ALL individuals DO Initialize() END FOR;
(2) FOR ALL individuals DO Evaluate()=>fitness END FOR;
(3) Sort(individuals according to fitness);
(4) IF terminal condition is met THEN STOP END IF;
(5) FOR ALL individuals DO
      SELECT Rand() OF
   CASE a DO Mutate()=> new_individuals;
   CASE b DO Symmetric_crossover(i-th, (i+1)th individuals) => new_individuals;
   CASE c DO One_point_crossover(i-th, (i+1)th individuals) => new_individuals;
   CASE d DO Re-generating() => new_individuals;
   END SELECT;
END FOR;
(6) FOR ALL new_individuals DO
   New ES_algorithm_object with related ES_individuals and ES_fitness arrays
//for each GPA individual new independent parameter optimizer is created
FOR ALL ES_individuals DO Initialize() END FOR;
    Evaluate() => ES_fitness;
FOR ALL ES_cycles DO
   FOR ALL ES_individuals DO
      Evaluate() => ES_fitness;
   END FOR;
   FOR ALL ES_individuals DO
         Intelligent_crossover() => new_ES_individuals
         Evaluate() => new_ES_fitness;
   END FOR;
   FOR ALL ES_individuals DO
         IF new_ES_fitness < ES_fitness THEN
            ES_individual = new_ES_individual;
            ES_fitness = new_ES_fitness;
      END IF;
   END FOR;
   Sort(ES_individuals, ES_fitness);
END FOR;
new_individual = ES_individual[0]; new_fitness = ES_fitness[0];
END FOR;
(7) FOR ALL individuals DO IF new_fitness < fitness THEN
      individual = new_individual;
      fitness = new_fitness;
      END IF;
    END FOR;
(8) GOTO 3);
```

Algorithm 1: Used GPA-ES algorithm structure.

GPA function set. Herein the presented experiments are based on ungrouped data measured with 1-minute resolution containing the first 20 values (delays 1, 2, ..., 20 minutes).

3. Results and Discussion

The GPA-ES algorithm worked with a maximal limit of 10000 GPA evolutionary cycles and 200 ES cycles in each GPA one, with 100 GPA individuals and 200 individuals in each ES population. This algorithm was written in the *C++* language and it was executed as a single-thread task on a single core of the 24 HT cores Intel Xeon processor. Execution times were between 80210 seconds for Sp trains and 274081 seconds for Ex trains due to stochastic character of the algorithm and resulting function shape.

The experiment majority tend to the presence of power function '^' in the resulting function. Add function '+' is the more useful (the more frequently present) than the divide function '/'. Also, the multiplication function '*' was frequently used. Factorial function '!' was never present in discovered functions. Thus, in future experiments, it can be left out and large data vectors can be used for probability density function learning.

The differences between passenger and freight train behaviour were observed rather in the coefficient magnitudes than in the structure of the discovered functions. All functions described in Table 2 are similar to FF of exponential distribution, but not identical.

Figures 2 and 3 provide examples of results for some train categories and allows comparing of discovered FF with original data frequencies.

Table 2: The FFs discovered by the genetic programming algorithm and residual errors of this estimation.

Train category	Train category delay frequency function	Sum of error squares
EC	$2(x-1)+7020.79*0.497426^{x-1}$	2.40 E+05
Ex	$(7610.04+(x-1)^2)((2x)^{-0.276312(x-1)})$	1.83 E+05
IC	$4966.33(0.686090^x)$	1.17 E+05
Lv	$(-3005.93+(3718.59(x-1)))(0.491252^{x-1})$	3.63 E+05
Mn	$(1463.47+3(x-1))*(0.846728^{x-1})$	1.49 E+05
Nex	$5010.84(x^{-0.130751(x-1)})$	5.96 E+05
Os	$134069(x(x-1))^{-0.188082(x-1)}$	4.72 E+07
Pn	$(10335.4(x-1))(0.51124^{(x-1)})$	1.03 E+07
R	$53062.1*506948^{-0.00965108(x-1)}$	4.31 E+07
Sp	$(x^{-0.461526(x-1)})(11695.3+x+(x-1)^2)$	3.69 E+05

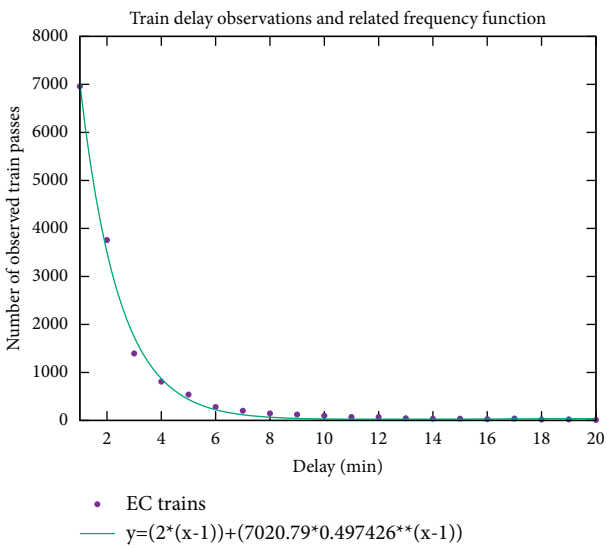


Figure 2: Example of an EC train category delay data histogram and related frequency function drawn as a continuous one to demonstrate its shape.

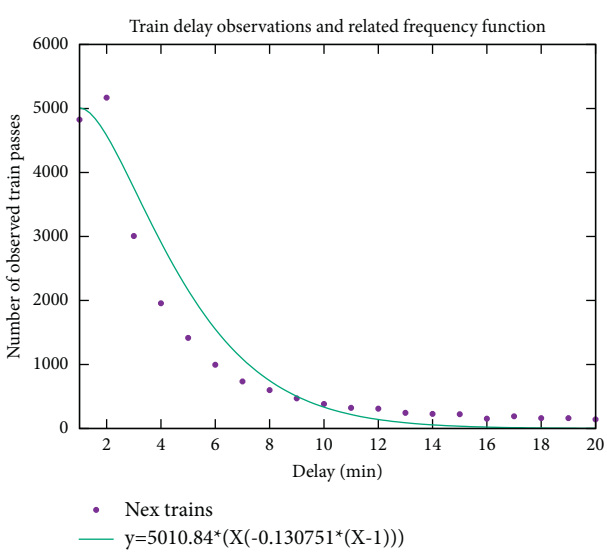


Figure 3: Example of Nex train category delay data histogram and related frequency function drawn as a continuous one to demonstrate its shape.

4. Conclusions

The presented study was devoted to the estimation of data distribution class by the application of the hybrid evolutionary algorithm GPA-ES and symbolic regression of frequency function equations.

The experiments described in this paper confirmed the exponential like distribution of train delays on the basis of FF regression from Czech railway network data. The evaluated amount of data was 6 GB and they were preprocessed by the Apache Spark application written in the *Python* language.

Founded FFs are similar to ones related to exponential distribution, but not identical.

The GPA-ES algorithm is capable to discover algebraic relations in applicable form for extremely small data vectors of 20 elements [24, 25].

Data Availability

The csv data used to support the findings of this study were supplied by the state company: Správa železnic, s.o. (Rail Infrastructure Administration, state organization) under license and are subject of commercial confidentiality, so they cannot be made freely available. Address of the data owner is Správa železnic, s.o., Dlážděná 1003/7, PSČ 110 00 Praha 1, The Czech Republic.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

The work was supported from ERDF/ESF "Cooperation in Applied Research between the University of Pardubice and Companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans)" (No. CZ.02.1.01/0.0/0.0/17_049/0008394).

References

[1] O. A. Nielsen, O. Landex, and R. D. Frederiksen, "Passenger delay models for rail networks, schedule-based modeling of transportation networks," in *Schedule-Based Modeling of*

Transportation Networks: Theory and Applications, pp. 1–23, Springer, Boston, MA, USA, 2009.

[2] A. Higgins, E. Kozan, and L. Ferreira, "Modelling delay risks associated with train schedules," *Transportation Planning and Technology*, vol. 19, no. 2, pp. 89–108, 1995.

[3] A. Kavicka, M. Bazant, and V. Zahorova, *Statistic Analysis of Data about Train Delays in the Railway Network*, United Progressive Alliance, Pardubice, Czech Republic, 2014.

[4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.

[5] C. Ryan, J. Collins, and M. O. Neill, "Grammatical evolution: evolving programs for an arbitrary language," in *Genetic Programming*, W. Banzhaf, R. Poli, M. Schoenauer, and T. C. Fogary, Eds., Springer, Berlin, Germany, pp. 83–96, Lecture Notes in Computer Science, 1998.

[6] C. Ferreira, "Gene expression programming: a new adaptive algorithm for solving problems," 2001, https://arxiv.org/abs/cs/0102027.

[7] J. F. Miller and P. Thomson, J. Miller, Edited by R. Poli and W. Banzhaf, Eds., "Cartesian genetic programming," in *Genetic Programming*, P. Nordin and T. C. Fogarty, Eds., Springer, Berlin, Germany, pp. 121–132, Lecture Notes in Computer Science, 2000.

[8] D. Jackson, "Single node genetic programming on problems with side effects," in *Parallel Problem Solving from Nature - PPSN XII*, C. A. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone, Eds., Springer, Berlin, Germany, pp. 327–336, Lecture Notes in Computer Science, 2012.

[9] J. Kubalík, E. Alibekov, J. Žegklitz, and R. Babuška, "Hybrid single node genetic programming for symbolic regression," in *Transactions on Computational Collective Intelligence XXIV*, vol. 9770, pp. 61–82, Springer, Berlin, Germany, 2016.

[10] T. Brandejsky and I. Zelinka, Edited by I. Zelinka, O. E. Rössler, and V. Snášel, Eds., "Specific behaviour of GPA-ES evolutionary system observed in deterministic chaos regression," in *Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems*, A. Abraham and E. S. Corchado, Eds., Springer, Berlin, Germany, pp. 73–81, Advances in Intelligent Systems and Computing, 2013.

[11] J. Frohlich and C. Hafner, "Extended and generalized genetic programming for function analysis," *Submitted to Journal of Evolutionary Computation*, 1996.

[12] T. L. Lew, A. B. Spencer, F. Scarpa, K. Worden, A. Rutherford, and F. Hemez, "Identification of response surface models using genetic programming," *Mechanical Systems and Signal Processing*, vol. 20, no. 8, pp. 1819–1831, 2006.

[13] M. Virgolin, T. Alderliesten, A. Bel, C. Witteveen, and P. A. N. Bosman, "Symbolic regression and feature construction with GP-GOMEAapplied to radiotherapy dose reconstruction of childhood cancer survivors," in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*, H. E. Aguirre and K. Takadama, Eds., pp. 1395–1402, ACM, Kyoto, Japan, July 2018.

[14] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, no. 5923, pp. 81–85, 2009, https://science.sciencemag.org/content/324/5923/81.

[15] S. Luke and L. Panait, "A comparison of bloat control methods for genetic programming," *Evolutionary Computation*, vol. 14, no. 3, pp. 309–344, 2006.

[16] W. B. Langdon and B. F. Buxton, "Genetic programming for mining DNA chip data from cancer patients," *Genetic Programming and Evolvable Machines*, vol. 5, no. 3, pp. 251–257, 2004.

[17] J. Li, K. Cheng, S. Wang et al., "Feature selection: a data perspective," *ACM Computing Surveys*, vol. 50, no. 6, Article ID 94, 2017.

[18] C. Hafner and J. Frohlich, "Generalized function analysis using hybrid evolutionary algorithms," in *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, USA, July 1996.

[19] G. R. Raidl, "A hybrid GP approach for numerically robust symbolic regression," in *Proceedings of the 3rd Annual Genetic Programming Conference*, J. Koza, Ed., pp. 323–328, Morgan Kaufmann, San Francisco, CA, USA, https://www.ac.tuwien.ac.at/files/pub/raidl.pdf.

[20] B. McKay, "Using a tree structured genetic algorithm to perform symbolic regression," *IET Conference Proceedings*, no. 5, pp. 487–492, 1995, https://digital-library.theiet.org/content/conferences/10.1049/cp_19951096.

[21] F. Qi, Y. Ma, X. Liu, and G. Ji, "A hybrid genetic programming with particle swarm optimization," in *Advances in Swarm Intelligence*, T. Ying, S. Yuhui, and M. Hongwei, Eds., Springer, Berlin, Germany, pp. 11–18, Lecture Notes in Computer Science, 2013.

[22] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference*, Perth, Australia, December 1995.

[23] Q. Chen, B. Xue, L. Shang, and M. Zhang, "Improving generalisation of genetic programming for symbolic regression with structural risk minimisation," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '16)*, pp. 709–716, Association for Computing Machinery, New York, NY, USA, July 2016.

[24] T. Brandejsky, "Multi-layered evolutionary system suitable to symbolic model regression," in *Proceedings of the NAUN/IEEE.AM International Conferences. 2nd International Conference on Applied Informatics and Computing Theory*, pp. 222–225, WSEAS Press, Praha, Athens, 2011.

[25] T. Brandejsky, "Symbolic regression of deterministic chaos," in *Proceedings of the 17th International Conference on Soft Computing (MENDEL 2011)*, pp. 90–93, VUT v Brně, Brno, Czech Republic, June 2011, ISSN 1803-3814.

[26] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*, Lulu Enterprises UK Ltd, London, UK, 2018.

[27] T. El-mihoub, A. Hopgood, L. Nolle, and A. Battersby, "Hybrid genetic algorithms: a review," *Engineering Letters*, vol. 3, no. 2, 2006.

[28] R. Poli, W. B. Langdon, P. K. Chawdhry, R. Roy, and R. K. Pant, *Soft Computing in Engineering Design and Manufacturiing*, pp. 180–189, Springer, London, UK, 1998.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 8840516, 22 pages
https://doi.org/10.1155/2021/8840516

Research Article

Dynamic Automated Search of Shunting Routes within Mesoscopic Rail-Traffic Simulators

Antonin Kavička [1] and Pavel Krýže [2]

[1] *Faculty of Electrical Engineering and Informatics, University of Pardubice, Studentska 95, 532 10 Pardubice, Czech Republic*
[2] *Správa železnic, s. o. (Railway Infrastructure Administration, State Organization), Dlážděná 1003/7, 110 00 Prague, Czech Republic*

Correspondence should be addressed to Antonin Kavička; antonin.kavicka@gmail.com

Received 11 September 2020; Revised 16 October 2020; Accepted 18 March 2021; Published 5 April 2021

Academic Editor: Avinash Unnikrishnan

Software tools using computer simulations are frequently used in the research and optimization of railway transport systems. Such simulations serve to examine different railway traffic scenarios (which typically reflect different timetables and railway infrastructure configurations). During the simulation experiments, it is necessary, among other things, to solve tasks related to the determination of track routes along which individual trains or parts of train sets are moved. Many simulation tools require the basic and alternative permissible track routes to be manually specified before starting the simulations, which is a relatively tedious and time-consuming process. Classical graph algorithms cannot be applied to solve the problem of automatic calculation of the routes because they are unable to take into account the length of the object being moved or recognise changes in the direction of its movement. This article presents original innovative algorithms focused on automated dynamic search of track routes (applying an appropriate optimization criterion), which is performed during simulation experiments within simulators working at the mesoscopic level of detail. The algorithms are based on a mathematical model (represented by a specifically designed weighted digraph) that appropriately reflects the actual track infrastructure. The dynamic calculation of each specific track route for a train or a group of railway vehicles considers both the total train set length and the current railway infrastructure occupancy, including blocked parts of the infrastructure due to intervention of the interlocking system. In addition, the places where the train set movement direction is changed can be identified on each route found. Applications of the algorithms and of the mathematical model of the track layout are demonstrated on a model track infrastructure.

1. Introduction

The research method of computer simulation is frequently used to examine and optimise the operation of railway systems. In this context, software simulators, or simulation tools, specialized to simulate railway traffic are used with advantages. Such tools require decisions associated with various types of operational tasks/situations to be automatically taken during the simulation experiments. Examples include decision on the assignment of alternative platform tracks to arriving delayed trains, search for admissible train routes available for train set shunting in a currently occupied trackage, and train selection (from a set of more than one waiting train) for permission to enter a specific line track.

This article is mainly devoted to automated dynamic computations of track routes required to relocate rail vehicles on a rail infrastructure, intended for inclusion into traffic simulations. Addressing this problem was primarily motivated by efforts to improve the capabilities of current rail-traffic simulators, which often enable static specification of the above track routes to be made only manually (before starting the individual simulation experiments). The possibility of including dynamic computations of track routes for relocating objects (having a given length) into the run of a simulation experiment has a high potential to facilitate (and shorten) the process of simulation experiment parameterization. The dynamic track route computation procedures are based on original algorithms that clearly

represent new (as yet unused) solutions in the rail-traffic simulation domain.

For a simulator to be classified as a credible tool for practical uses, the traffic and decision processes used within the simulation experiments must be such that the simulation should approach the real railway system operation as best as possible. In other words, it is very important that the automated solutions implemented in the simulator should be based on suitable models and methods providing results applicable in real traffic situations.

The present article describes an innovative automated solution of one type of operational problems, namely, determination of the train route topology or shunting route topology for relocation of a train/train set within the railway infrastructure. Information for the specification of the relocation encompasses the relocation object length, starting and final positions (tracks), and current railway yard occupation. The solution has been developed for use in rail-traffic simulators working at the mesoscopic level of detail.

2. Literature Review

The operation of railway systems is examined and optimized by employing a number of different methods and techniques that are supported by various software tools. The main goal of such optimizations is to find such solutions which can be used to help ensure quality traffic. The term quality traffic can be interpreted (with some simplifications) as traffic that (i) basically (with minor deviations) follows the timetable and (ii) uses the service resources economically, i.e., means of transport (such as shunting locomotives), elements of infrastructure (on which rail vehicles are moved), and human resources (e.g., technical personnel working in the field of rail yards). A number of specific optimization tasks can be identified in the complex of provisions to ensure quality traffic, and each task may use suitable models of the infrastructure and models of the traffic and traffic management systems.

One of the important optimization tasks consists in solving the train routing problem. This extensive problem includes, for instance, the following partial tasks: rail line assignment to trains within the extensive areas of railway networks [1, 2]; coordinated assignment of track routes to more than one train in the railway station [3–6]; and identification of suitable track routes for rail vehicle shunting [7]. Among the relevant factors playing a role when solving such problems is the configuration of the specific infrastructure on which the traffic takes place. This implies that one should seek to select the best-fitting track infrastructure model and use the most suitable algorithms for each specific type of examination.

Mathematical structures of the graph type (and their specific implementations), which fall in the graph theory domain, are frequently used when describing a particular track infrastructure. Infrastructure models can be constructed both on directed graphs (digraphs) and on undirected graphs. Original track layout models built on the double vertex graph [8] and polar graph [7, 9] concepts make possible distinction between track segments of switches and station track segments.

Among typical tasks addressed by using graphs is the search for admissible (potentially shortest) track routes on which rail vehicles are shunted. The concept of the original Dijkstra's algorithm [10–12], which is primarily aimed at searching for the single-source shortest paths on a weighted directed graph, can be used with advantage for such purposes. Numerous modifications of this algorithm have also been used when examining railway systems [13–15].

One of the important methods used to investigate and optimise railway systems is computer simulation [16]. Different track layout models are used within software tools specialized to simulate rail traffic [7, 8, 14, 15]. Examples of relevant simulation tools in this domain include OpenTrack [17], RailSys [18], Villon [19], MesoRail [20], NEMO [21], and PULSim [22]. Such tools always apply the same level of detail—microscopic, mesoscopic, or macroscopic. On the contrary, different approaches can apply distinct levels of detail within different parts of a simulator—such simulations are referred to as multiscale simulations [23] or hybrid simulations [24, 25]. The existence of software platforms for combined traffic simulations, capable of examining interactions of different traffic modes (e.g., [26]), is also worth mentioning.

In order to work, the above software tools require data description of the rail infrastructure, which can be available in various formats of the configuration files. A standard exists for this purpose, viz. railML [27, 28] (open-source railway markup language). This standard defines a recommended format of the files for the exchange of data for railway applications. The RailTopoModel [29, 30] is also available: this is a logical object model designed for standardization of the representation of railway infrastructure-related data.

Continuation of that part of the research, the results of which were published in [7], was motivated by the need for specialized functions to be applied in the new simulation tool named MesoRail, serving rail-traffic simulations and working at the mesoscopic level of detail. Among such functions was automated computation of the topologies of the track routes on which relocation objects (such as trains and locomotives) will be moved within the track infrastructure model. For this, the appropriate original algorithms (which, however, have never been published so far) were redesigned and implemented. The algorithms were then tested with success within MesoRail for use in dynamic track route calculations. The track layout model used and the algorithms working on it make up original solutions that have never been used elsewhere (according to the available literature).

3. Models of the Rail Infrastructure

Automated dynamic search for train routes and shunting routes within the railway infrastructure uses a mathematical model, reflecting the track layout examined. The construction of such a model consists of 2 stages.

3.1. Primary Model. Stage 1 includes the creation of a primary model using an undirected edge-weighted graph as specified in Table 1. The edges in the graph represent the individual tracks (or their parts) as well as the track segments of the switches. The edges reflecting the tracks are referred to as destination edges, while the remaining edges are referred to as connecting edges.

This terminology mirrors the fact that tracks can be viewed as targets for rail vehicle relocation whereas switches and track crossings cannot. The graph vertices represent the contact points of the track segments reflected by the edges.

Where switches and crossings are modelled, their different types determining how they can be technically transited must be differentiated. Some examples of how different switches or crossings are represented in graph G_0 are illustrated in Figure 1. Admissible transits in Figure 1 are specified in Table 2.

Each edge in graph G_0 has a weight assigned, expressing the metric length of the track segment. An example of a primary model mirroring a demonstration railway yard (total track length: 3049 m) is shown in Figure 2.

When a specific rail vehicle relocation within the rail infrastructure is required, the starting and final positions are assumed to be associated with specific tracks (represented by appropriate edges in graph G_0). In order to make it possible to distinguish between the opposite track ends, either end of each edge in the graph can be assigned either the plus sign (+) or the minus sign (−). This enables us, for instance, to specify that end of the finish track through which the rail vehicle should reach the finish position. Such edge end labelling can also be interpreted as an alternative labelling of the vertex that is incident with the specific end of the specific edge. For example, the notation ${}^{-}e_5 = v_{12} = {}^{+}e_{20}$ can be used in Figure 2. The assignment of the signs to the opposite edge ends can be based on a rule (for instance, if the graph is placed in the two-dimensional coordinate system, then that track end having a lower value of coordinate x is assigned the plus sign, the opposite end, the minus sign).

The use of this marking was partly inspired by the polar graph concept [7, 9]. Each vertex in a graph of this type is composed of two poles—the plus pole and the minus pole—and each edge incident with a vertex can be classed in one of the following two categories: edges incident with the plus pole and edges incident with the minus pole of the vertex. This principle was loosely applied to the marking of the opposite ends of the undirected edges in graph G_0. Figure 2(b) also illustrates the graphical encapsulation of the groups of edges corresponding to the switch objects (designated with symbol S_i, $i = 1, \ldots, 6$).

3.2. Final Model. Stage 2 includes transformation of the primary model G_0 into the final model (Figure 3), which is represented by a specific weighted digraph G. The specification of this graph is presented in Table 3. If the transformation is used, an ordered pair of appropriate vertices ${}^{i}v_x \in V(G)$, $i = 1, 2$, is formed for each edge $e_x \in E(G_0)$ (that is, the bijective transformation function τ: $E(G_0) \longrightarrow \{[{}^{1}v_x, {}^{2}v_x] \mid {}^{1}v_x, {}^{2}v_x \in V(G), {}^{1}v_x \neq {}^{2}v_x, [{}^{1}v_x, {}^{2}v_x], [{}^{2}v_x, {}^{1}v_x] \notin E(G)\}$ is applied). The two vertices representing a track in the digraph G mirror the fact that the track can be entered/left in two opposite directions.

Furthermore, the vertices in digraph G can be categorised according to whether they can represent the relocation targets or not. Hence, we distinguish between destination vertices (represented by full circles in Figure 3) and connecting vertices (represented by empty circles).

Edges in the digraph G can be categorised as transit edges (shown as full lines in Figure 3) and reverse edges (shown as dashed lines). Transit edge expresses the possibility of transiting through the track modelled with a vertex (that represents a starting point of the transit edge) to adjacent tracks. Reverse edge expresses the fact that a train can change the direction of its motion (=reversal) on the track that is modelled by the starting vertex of the reverse edge.

Two transit edges representing the possibility of transiting between the tracks in 2 directions are constructed in graph G for each pair of adjacent edges (the edge construction methods are explained in Table 4). Reverse edges can start from destination vertices solely. Here, it is assumed that the whole relocation object stops on one track during the reversal operation. A situation where a train can occupy more than one track during reversal is not modelled in the digraph G. This simplification can be applied within simulators operating at the mesoscopic level of detail.

Applying the rule specified in Table 4, a reverse edge is constructed for each transit edge whose end vertex is a destination vertex. For example (referring to the graph in Figure 2), the reverse edge $[{}^{2}v_5, {}^{1}v_{18}]$ is formed to the transit edge $[{}^{2}v_{18}, {}^{2}v_5]$.

Digraph G is both vertex-weighted and edge-weighted, and the vacancies are preserved for the vertices. The following rules are applied to the weights and vacancies:

(i) The weights of the vertices in digraph G (expressed through function ω) represent the metric lengths of the track segments corresponding to the weights of the respective edges in graph G_0.

(ii) The weights of transit edges (expressed through function ε) in digraph G are identical with the weights of the vertices they proceed from. The weight of a specific transit edge represents the distance run by the relocation object by transiting the track that mirrors the starting vertex of that edge.

(iii) The weights of reverse edges (expressed through function ε) are identical for all those edges and do not attain fixed values: instead, they are set (to value L) always before running the computing algorithm searching for a given train or shunting route (for a relocation object whose length is L). This weight is interpreted so that if the relocation object stops on a certain track (corresponding to the starting vertex of a reverse edge) for a time and then continues by moving in the opposite direction, then the used part of the track is that part whose length is identical with that of the relocation object (i.e., L).

Table 1: Specification of the weighted undirected graph G_0—the primary model of the track infrastructure.

Symbols	Specifications
G_0	The weighted undirected graph (i) $G_0 = (V, E, \varphi, \varepsilon)$
$V(G_0)$	The set of vertices of the graph G_0 (i) $\lvert V(G_0)\rvert = n_0$
$E(G_0)$	The set of edges of the graph G_0 (i) $\lvert E(G_0)\rvert = m_0$ (ii) $E(G_0) = E_{\text{dest}}(G_0) \cup E_{\text{conn}}(G_0)$, $E_{\text{dest}}(G_0) \cap E_{\text{conn}}(G_0) = \varnothing$ (iii) The set $E_{\text{dest}}(G_0)$ contains destination edges (iv) The set $E_{\text{conn}}(G_0)$ contains connecting edges
φ	The incidence function related to the graph G_0 (i) φ: $E(G_0) \longrightarrow \{(v_x, v_y) \mid (v_x, v_y) \in V(G_0) \times V(G_0) \wedge v_x \neq v_y\}$
ε	The edge weight function related to the graph G_0 (i) ε: $E(G_0) \longrightarrow R^+$ (the set of positive real numbers)

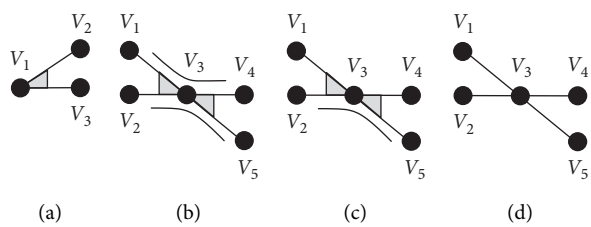


Figure 1: The models of switches and crossings applied within undirected graphs.

Table 2: Admissible transits through several kinds of switches and crossings.

The type of switch	Admissible transits
Switch (Figure 1(a))	$v_1 \longrightarrow v_2$, $v_2 \longrightarrow v_1$, $v_1 \longrightarrow v_3$, $v_3 \longrightarrow v_1$
Double slip switch (Figure 1(b))	$v_1 \longrightarrow v_5$, $v_5 \longrightarrow v_1$, $v_1 \longrightarrow v_4$, $v_4 \longrightarrow v_1$, $v_2 \longrightarrow v_4$, $v_4 \longrightarrow v_2$, $v_2 \longrightarrow v_5$, $v_5 \longrightarrow v_2$
Single slip switch (Figure 1(c))	$v_1 \longrightarrow v_5$, $v_5 \longrightarrow v_1$, $v_2 \longrightarrow v_4$, $v_4 \longrightarrow v_2$, $v_2 \longrightarrow v_5$, $v_5 \longrightarrow v_2$
Track crossing (Figure 1(d))	$v_1 \longrightarrow v_5$, $v_5 \longrightarrow v_1$, $v_2 \longrightarrow v_4$, $v_4 \longrightarrow v_2$

(iv) Furthermore, every vertex in digraph G has an available vacancy (expressed through function κ), expressing the metric free capacity of the respective track. Since each track in digraph G is modelled with a pair of vertices, the vacancies of the vertices express the metric lengths of the vacant parts of the track from the two opposite ends. Function κ can be used to express the rail infrastructure occupation by rail vehicles and also the blocking if its parts are unreachable due to the operation of the interlocking system (for instance, if a certain track x is blocked, then the equation $\kappa\,({}^1v_x) = \kappa\,({}^2v_x) = 0$ holds for the graph vertices ${}^1v_x, {}^2v_x \in V(G)$).

Examples of particular weight and vacancy values belonging to selected elements in digraph G are shown in the following (Example 1).

4. Algorithms Focused on Searching Train Routes

Now, after the final railway infrastructure model has been introduced, the algorithms for calculating train routes or shunting routes can be described. The search of the routes was based on the optimization principle consisting in minimisation of the distance run by the relocation object in question.

Dijkstra's algorithm [10, 11] for searching the shortest routes between the vertices on an edge-weighted graph was selected as the starting algorithm. However, it had to be appreciably modified for use in the calculation of the relocation trajectories for trains or train sets (having a defined length) on the railway infrastructure.

As mentioned earlier, the railway infrastructure was modelled by digraph G, specified in Table 3.

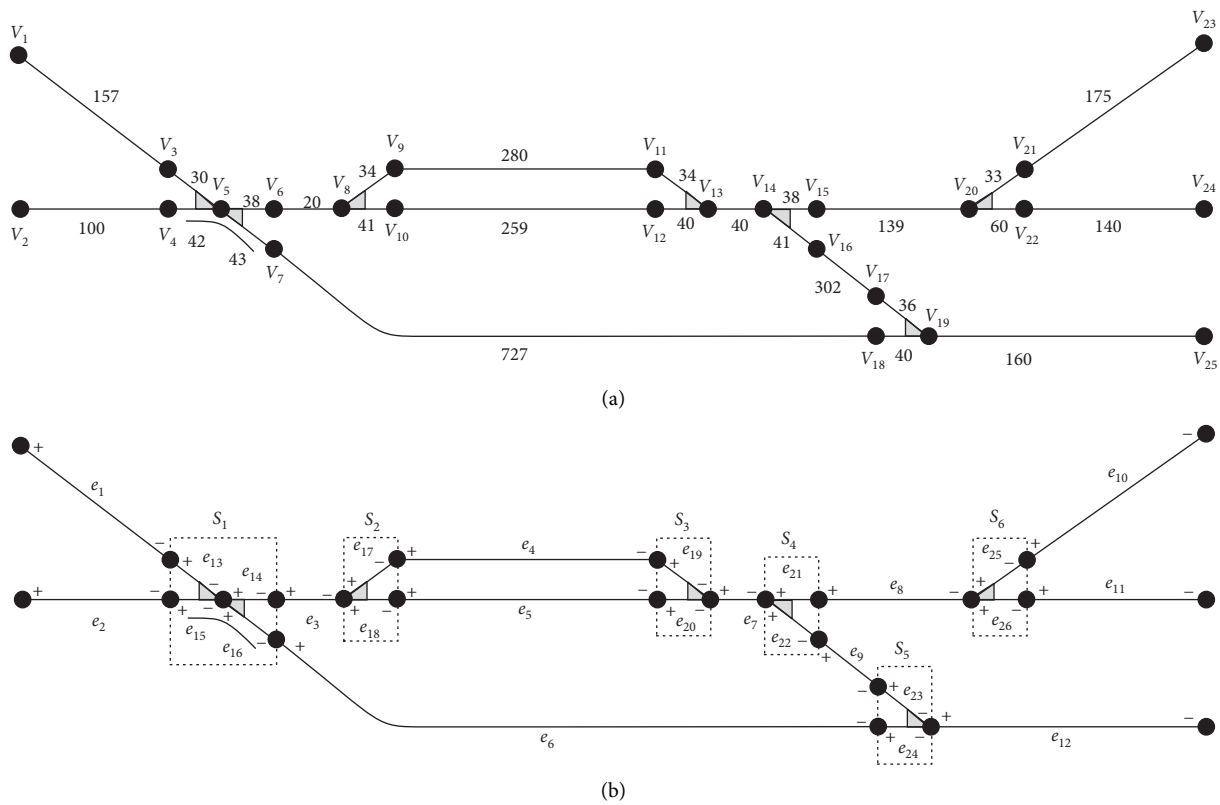


Figure 2: Prototype track infrastructure—a primary model based on an undirected graph G_0.

Now, additional auxiliary sets, row vectors, and subroutines/functions to be used in the algorithms must be specified in order to enable the algorithms to be formalized. The specifications are listed in Tables 5 and 6. The subroutines/functions utilise parameters applying the following convention: the symbol "↓" denotes an input parameter, symbol "↑" denotes an output parameter, and double symbol "↓↑" denotes an input-output parameter.

4.1. Basic Algorithm. The basic algorithm (formalized as Algorithm 1) searches for the shortest route in the track layout from a specific start vertex to a specific finish vertex. The start vertex represents the end of the start track transited by the train (whole length is L) when starting its relocation procedure, while the finish vertex represents the end of the finish track transited by the train when approaching the finish position.

The main function, *Shortest_Path*, calculates the shortest relocation route for an object (whose length is L), given the start vertex (from the set S_V) and finish vertex (from the set F_V). The topology of this route is available through the parameter *Topol*. For Algorithm 1, the sets S_V and F_V include one vertex each. Further modifications of this algorithm, however, will permit one or two vertices to be included in either set.

The following procedures are carried out within the subroutines/functions (called from the main function *Shortest_Path*).

First, the function *Start_Finish_Test* tests admissible combinations of the input parameter values for the route calculation. For the combination to be acceptable, the weight of the start vertex must not be lower than the relocation object length (L), and the finish vertex must have a sufficient vacancy. Furthermore, the Boolean expression ($x = y \wedge i \neq j$) must not hold true for the start vertex ${}^{i}v_x$ and finish vertex ${}^{j}v_y$ (the above expression describes a situation where the start and finish vertices refer to the same track from which the relocation procedure is started and which is entered by transiting the same end).

Furthermore, the function *General_Init* is used for initialisation of the marks of distances and the marks of predecessors (through vectors D and P) for all vertices in the graph. The distance marks of all vertices in the digraph G are set at the value d^{∞} (which is larger than the longest distance between any vertices in the digraph G). The marks of predecessors are set at none for all vertices in the digraph G. The set T_V of temporarily marked vertices and the set U_V of ultimately marked finish vertices are initialised as empty sets.

Additional initialisation procedures associated with the start and finish vertices are subsequently run (through the function *Start_Finish_Init*). The start vertex is assigned a distance mark whose value is 0 (i.e., the length of the current shortest route is zero for the start vertex). Now, the set of forbidden vertices is constructed. The vertices from this set must not lie on the shortest route being sought. When the concept of forbidden vertices is applied, the relocation

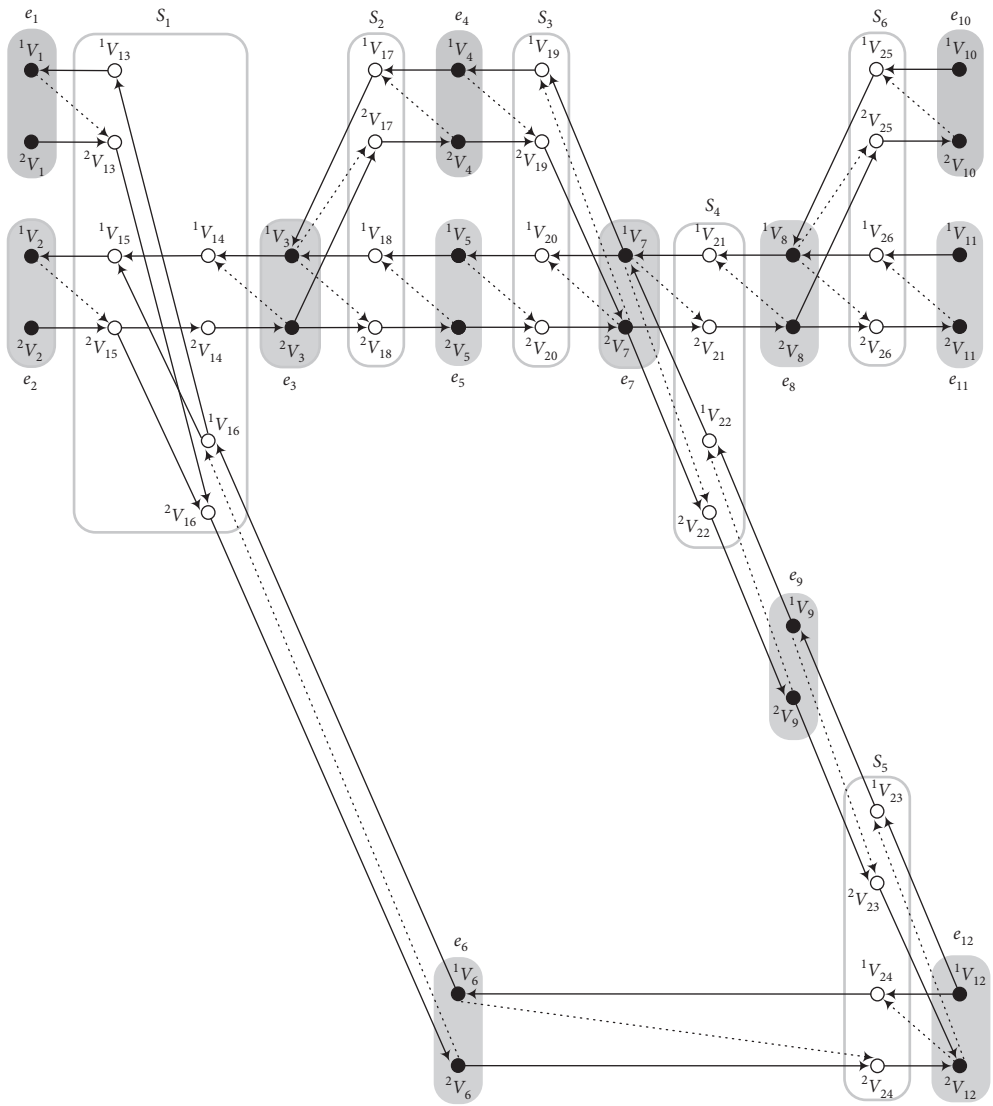


Figure 3: The final infrastructure model based on digraph G.

trajectory must not pass through vertices describing those ends of tracks over which it is inadmissible to reach the finish track (with respect to the input requirements for relocation).

Next, the admissible successors of the start vertices that can be reached through transit edges are remarked (through the function *Start_Mark*). These vertices-successors are assigned (i) distance marks with a value of 0 (through vector D) and (ii) marks of predecessor (in vector P) referring to the start vertex. The remarked vertices are also inserted into the set T_V.

The complex of initialisation procedures is followed by a computation cycle involving selection of a vertex (through the function *Vert_Select*) for processing in the current iteration step and the marking of all its admissible successors which is potentially changed (through the function *Succ_Mark*). This computation cycle is terminated if all the appropriate finish vertices have been ultimately marked or if T_V—the set of temporarily marked vertices—is empty.

The *Vert_Select* function always selects that vertex ${}^k v_z$ from the set T_V that has the lowest value of the distance mark ${}^k d_z$. The *Succ_Mark* function changes the marking of the accessible successors of the selected vertex ${}^k v_z$ if those successors (i) have an available adequate vacancy; (ii) they are not members of the set of forbidden vertices; and (iii) the current values of their distance marks are higher than the new values mirroring their reaching through another route (via vertex v_z). Moreover, the Boolean expression $(\kappa({}^k v_z) = \omega({}^k v_z))$ must hold true for the transit successors to be accessible from the vertex ${}^k v_z$. This expresses the condition that the track to which the vertex ${}^k v_z$ refers must be fully available for transiting. This means that the vacancy of the vertex ${}^k v_z$ must be equal to the weight of that vertex.

If the shortest route has been found, its topology is obtained through the function *Get_Path*. The topology is constructed stepwise from the finish vertex to the start vertex by using marks about the vertex predecessors (saved in the row vector P) on the shortest route.

Table 3: Specification of the weighted digraph G—the final model of the track infrastructure.

Symbols	Specifications
G	The weighted digraph (i) $G = (V, E, \varphi, \omega, \varepsilon, \kappa)$ (ii) The digraph G represents a result related to the transformation of the relevant undirected graph G_0
$V(G)$	The set of vertices of the digraph G (i) $V(G) = \{{}^k v_z \mid z = 1, \ldots, n\backslash 2,\ k = 1, 2\}$, $\lvert V(G)\rvert = n$ (ii) $V(G) = V_{\text{dest}}(G) \cup V_{\text{conn}}(G)$, $V_{\text{dest}}(G) \cap V_{\text{conn}}(G) = \varnothing$ (iii) The set $V_{\text{dest}}(G)$ contains destination vertices of the digraph G (iv) The set $V_{\text{conn}}(G)$ contains connecting vertices of the digraph G Note: symbol "\" denotes an integer division
$E(G)$	The set of directed edges of the digraph G (i) $\lvert E(G)\rvert = m$ (ii) $E(G) = E_{\text{trans}}(G) \cup E_{\text{rev}}(G)$, $E_{\text{trans}}(G) \cap E_{\text{rev}}(G) = \varnothing$ (iii) The set $E_{\text{trans}}(G)$ contains transit edges (iv) The set $E_{\text{rev}}(G)$ contains reverse edges
φ	The incidence function related to the digraph G (i) φ: $E(G) \longrightarrow \{[{}^i v_x, {}^j v_y] \mid [{}^i v_x, {}^j v_y] \in V(G) \times V(G) \wedge {}^i v_x \neq {}^j v_y;\ x, y \in <1, \ldots, n\backslash 2>;\ i, j \in \{1, 2\}\}$
${}^{\text{out}}V({}^k v_z)$	The set of successors of the vertex ${}^k v_z$ (i) ${}^{\text{out}}V({}^k v_z) = \{{}^l v_s \mid [{}^k v_z, {}^l v_s] \in E(G)\}$, ${}^{\text{out}}V({}^k v_z) \subset V(G)$ (ii) ${}^{\text{out}}V({}^k v_z) = {}^{\text{out}}V_{\text{trans}}({}^k v_z) \cup {}^{\text{out}}V_{\text{rev}}({}^k v_z)$, ${}^{\text{out}}V_{\text{trans}}({}^k v_z) \cap {}^{\text{out}}V_{\text{rev}}({}^k v_z) = \varnothing$ (iii) The set of transit successors of the vertex ${}^k v_z$ ${}^{\text{out}}V_{\text{trans}}({}^k v_z) = \{{}^l v_s \mid [{}^k v_z, {}^l v_s] \in E_{\text{trans}}(G)\}$ (iv) The set of reverse successors of the vertex ${}^k v_z$ ${}^{\text{out}}V_{\text{rev}}({}^k v_z) = \{{}^l v_s \mid [{}^k v_z, {}^l v_s] \in E_{\text{rev}}(G)\}$
ω	The vertex weight function related to the digraph G (i) ω: $V(G) \longrightarrow R^+$ (ii) $\forall\ {}^1 v_z, {}^2 v_z \in V(G)$ $(z = 1, \ldots, n\backslash 2)$: $\omega({}^1 v_z) = \omega({}^2 v_z)$
ε	The edge weight function related to the digraph G (i) ε: $E(G) \longrightarrow R^+$ (ii) $\forall\ [{}^i v_x, {}^j v_y] \in E_{\text{trans}}$: $\varepsilon([{}^i v_x, {}^j v_y]) = \omega({}^i v_x)$ (iii) $\forall\ [{}^i v_x, {}^j v_y] \in E_{\text{rev}}$: $\varepsilon([{}^i v_x, {}^j v_y]) = L$, where L represents a parameter value reflecting the length of a relevant relocation object
κ	The vertex vacancy function (vacant capacity function) (i) κ: $V(G) \longrightarrow R_0^+$ (ii) If a track segment (reflected by a vertex ${}^k v_z \in V(G)$) is completely vacant, then $\kappa({}^k v_z) = \omega({}^k v_z)$

Table 4: Construction of edges belonging to the digraph G transformed from a related graph G_0.

Adjacent edges $\varphi(e_x) \cap \varphi(e_y) \neq \varnothing$ $e_x, e_y \in E(G_0)$, $e_x \neq e_y$	Constructed transit edges $E_{\text{trans}}(G)$	Constructed reverse edges	
		$E_{\text{rev}}(G)$	Only if
${}^+e_x = {}^-e_y$	$[{}^1 v_x, {}^1 v_y]$, $[{}^2 v_y, {}^2 v_x]$	$[{}^1 v_y, {}^2 v_x]$ $[{}^2 v_x, {}^1 v_y]$	${}^1 v_y \in V_{\text{dest}}(G)$ ${}^2 v_x \in V_{dest}(G)$
${}^-e_x = {}^+e_y$	$[{}^2 v_x, {}^2 v_y]$, $[{}^1 v_y, {}^1 v_x]$	$[{}^2 v_y, {}^1 v_x]$ $[{}^1 v_x, {}^2 v_y]$	${}^2 v_y \in V_{\text{dest}}(G)$ ${}^1 v_x \in V_{\text{dest}}(G)$
${}^+e_x = {}^+e_y$	$[{}^1 v_x, {}^2 v_y]$, $[{}^1 v_y, {}^2 v_x]$	$[{}^2 v_y, {}^2 v_x]$ $[{}^2 v_x, {}^2 v_y]$	${}^2 v_y \in V_{\text{dest}}(G)$ ${}^2 v_x \in V_{\text{dest}}(G)$
${}^+e_x = {}^-e_y$	$[{}^2 v_x, {}^1 v_y]$, $[{}^2 v_y, {}^1 v_x]$	$[{}^1 v_y, {}^1 v_x]$ $[{}^1 v_x, {}^1 v_y]$	${}^1 v_y \in V_{\text{dest}}(G)$ ${}^1 v_x \in V_{\text{dest}}(G)$

At this point, the following comments should be added concerning the implementation of the above algorithm, transformation of its results, and an alternative view upon the start and finish points of the relocation trajectories:

(i) In view of the nature of the algorithm, a data structure called forward star [31] was used for implementation of the digraph G. The forward star structure consists of two basic arrays: a primary array that stores information on the graph vertices and a secondary array that stores information on the directed edges that start from the vertices (saved in the primary array). The secondary array is sorted by the starting edge vertices. This makes it possible for each vertex from the primary array to rapidly access (through its reference to the secondary array) the group of all its successors (which are always stored in the continuous part of the secondary array). For an effective work with the set T_V, the Fibonacci heap [11], as a very efficient realization of a priority queue, was chosen for its implementation. The

Table 5: Specifications of auxiliary sets and row vectors.

Symbols	Specifications
S_V	The set of start vertices (i) $S_V \subset V(G)$
F_V	The set of finish vertices (i) $F_V \subset V(G)$
X_V	The set of forbidden vertices (i) $X_V \subset V(G)$
T_V	The set of temporarily marked vertices (i) $T_V \subset V(G)$
U_V	The set of ultimately marked finish vertices (i) $U_V \subset V(G)$
D	The row vector of distances (i) $D = \|\|^k d_z\|\|$, ${}^k d_z \in R_0^+$, $z = 1, \ldots, n\backslash 2$, $k = 1, 2$ (ii) Each vertex ${}^k v_z \in V(G)$ is tagged by a mark ${}^k d_z$, which expresses the length of the currently detected shortest path from a vertex ${}^i v_x \in S_V$ to the vertex ${}^k v_z$ (iii) If the above path to the vertex ${}^k v_z \in V(G)$ does not exist, then ${}^k d_z = d^\infty$ ($d^\infty \in R_0^+$, and it is equal to the value that is greater than the length of the longest admissible path in the graph G) Note: symbol "\" denotes an integer division
P	The row vector of predecessors (i) $P = \|\|^k p_z\|\|$, ${}^k p_z \in V(G) \cup \{none\}$, $z = 1, \ldots, n\backslash 2$, $k = 1, 2$ (ii) Each vertex ${}^k v_z \in V(G)$ is tagged by a mark ${}^k p_z$, which corresponds to a predecessor of the vertex ${}^k v_z$ on the currently detected shortest path from a vertex ${}^i v_x \in S_V$ to the vertex ${}^k v_z$ (iii) If the above path to the vertex ${}^k v_z \in V(G)$ does not exist, then ${}^k p_z = none$ (the symbol *none* expresses the nonexistence of a relevant predecessor with regard to the vertex ${}^k v_z$)
Seq	The linearly ordered set of the shortest path topology (i) $Seq = \{[i, {}^k v_z] \mid i = 0, \ldots, q-1, {}^k v_z \in V(G), z \in \langle 1, \ldots, n\backslash 2\rangle, k \in \{1, 2\}, q = \|Seq\|\}$ (ii) The element $[0, {}^k v_z]$ represents a finish vertex (${}^k v_z \in F_V$) and the element $[q-1, {}^k v_z]$ a start vertex (${}^k v_z \in S_V$) of the shortest path

priorities of the elements (or vertices in the digraph G) expressed the values of the respective marks from the row vector D (the higher the priority of an element, the lower the value of its distance mark).

(ii) The configuration of the railway infrastructure can be specified for the needs of the MesoRail simulator [20], for example, within the TrackEd editor [24]. This editor stores infrastructure description in the XML format, which uses templates inspired by the railML standard [27, 28].

(iii) For the implementation approaches used (applying Dijkstra's algorithm concept), the asymptotical computational complexity of Algorithm 1 can be specified as $O(m + n \log n)$ [12], where $m = |E(G)|$ and $n = |V(G)|$. Nevertheless, for typical track routes that are not searched for long distances in real operational conditions, the relevant computations (within rail-traffic simulators) are concentrated on subgraphs (of the digraph G) that are usually not very extensive.

(iv) The result of Algorithm 1 computations for the shortest route found in the digraph G can be reversely transformed into the primary model, that is, into the undirected graph G_0. This means that the shortest route in the digraph G corresponds to the shortest walk in the graph G_0. The walks are outlined graphically in Figures 4–6, illustrating the solution of Examples 1–3 (see later).

(v) In standard parameterisation of the *Shortest_Path* function within Algorithm 1, the start and finish positions of the relocation procedure are assumed to be represented by vertices in the digraph G. This, in fact, means that the relocation object (whose length is L) is located "precisely" at the end of the respective track both at the beginning and at the end of the relocation operation. If it is required that the relocation operation starts and/or finishes on another part of the track, then Algorithm 1 must be slightly modified. The modification will encompass both the marking of the start vertex (and potentially its transit successors as well) and the ultimate marking of the finish vertex. The values of the distance marks assigned to the vertices within the *Start_Finish_Init* and *Start_Mark* functions will be set at a value different from 0 (denoted, e.g., $reloc_1$) describing the metric length of relocation to the postulated end of the start track. On the contrary, the value of the ultimate mark of the finish vertex of the relocation (denoted, for instance, *val*, and indicating the length of the shortest route found) will not describe the length of the relocation object trajectory precisely (denoted, for instance, *dist*). In fact, the length of the trajectory can be obtained as $dist = val + reloc_2$, where $reloc_2$ refers to the metric length of relocation from the postulated end of the finish track to the end position on that track.

Table 6: Specifications of subprograms/functions.

Subprograms/functions	Specifications
Shortest_Path($\downarrow S_V$, $\downarrow F_V$, $\downarrow L$, $\uparrow Topol$)	The calculation of the shortest path (the graph G) S_V (the set of start vertices) F_V (the set of finish vertices) L (the length of the relocation object) *Topol* (the topology of the shortest path)
Start_Finish_Test($\downarrow S_V$, $\downarrow F_V$, $\downarrow L$, $\uparrow\downarrow okay$)	The test of correctness of start and finish vertices S_V (the set of start vertices) F_V (the set of finish vertices) L (the length of the relocation object) *okay* (the test result)
General_Init()	The execution of general initialisation activities The initialisation of the sets T_V and U_V The initialisation of the row vectors P and D
Start_Finish_Init($\downarrow S_V$, $\downarrow F_V$)	The initialisation activities reflecting start vertices and finish vertices The initialisation of the set X_V The initialisation of the marks ${}^i d_x$ (${}^i d_x = 0$) related to the start vertices (${}^i v_x \in S_V$)
Start_Mark($\downarrow S_V$,$\downarrow L$)	The initialisation activities focused on markings of the successors of start vertices The update of the set T_V and the row vectors P and D
Vert_Select($\uparrow {}^k v_z$)	The selection of the vertex ${}^k v_z$ (with the current minimal distance mark) for the processing in the next step of the algorithm The vertex ${}^k v_z$ is selected/removed from the set T_V The set U_V is potentially updated
Succ_Mark($\downarrow {}^k v_z$, $\downarrow L$)	The potential execution of admissibly rewriting the marks belonging to all successors of the vertex ${}^k v_z$ The potential update of the set T_V and the row vectors P and D
Get_Path($\uparrow\downarrow Seq$)	The delivery of the topology related to the found shortest path *Seq* represents a sequence of vertices
Get_Indexes($\downarrow {}^i v_x$, $\uparrow i$, $\uparrow x$)	The auxiliary function delivering indexes of a vertex ${}^i v_x$
Min_Dist($\downarrow A$, $\uparrow {}^k v_z$)	The auxiliary function identifying a vertex ${}^k v_z$ (within a set A) associated with the minimum value of the relevant mark ${}^k d_z$
Try_Change_Mark($\downarrow {}^k v_z$, $\downarrow {}^l v_s$)	The auxiliary function potentially rewriting the marks belonging to the successor (the vertex ${}^l v_s$) of the vertex ${}^k v_z$ The potential update of the set T_V and the row vectors P and D
Predecessor($\downarrow {}^k v_z$, $\uparrow l$, $\uparrow t$)	The auxiliary function delivering indexes related to a predecessor (${}^l v_t$) of the vertex ${}^k v_z$ (on the relevant shortest path)

4.2. Examples of Using the Basic Algorithm. A few examples of the application of the basic algorithm (Algorithm 1) to the computation of various relocation actions are shown in the following. The first example concerns the transfer of a complete train between two station tracks.

Example 1. Computation of a shunting route topology for the train relocation ($L = 120$).

Figure 4 shows two models of a demonstration railway yard represented by graphs G and G_0. There is one train 120 m long (relocation object O_1) on track #5 (reflected by edge e_5 in graph G_0) of the yard. The train stands at the track end, referred to as ${}^- e_5$. The requirement is to find a route for transferring the object O_1 onto track #4 (represented by edge e_4 in graph G_0). The object O_1 should leave track #5 via the track end referred to as ${}^- e_5$ and enter track #4 via its end referred to as ${}^- e_4$. This setup reflects a situation where track #5 should be made available for other trains and the train in question should be transferred to track #4, so it can leave the yard in the direction to a rail line accessible via track #11.

Parameterization of the algorithm for the track route computation (shortest route in the digraph G) is as follows:

(i) $S_V = \{{}^2 v_5\}$, $F_V = \{{}^1 v_4\}$, and $L = 120$

The weights are given for selected vertices and edges of the relevant graphs, and current vacancies are quantified for selected vertices in the digraph G:

(i) $\omega\ ({}^1 v_4) = \omega\ ({}^2 v_4) = 280$, $\kappa\ ({}^1 v_4) = \kappa\ ({}^2 v_4) = 280$, ${}^1 v_4$, ${}^2 v_4 \in V(G)$, $\varepsilon\ (e_4) = 280$, $e_4 \in E(G_0)$

(ii) $\omega\ ({}^1 v_5) = \omega\ ({}^2 v_5) = 259$, $\kappa\ ({}^1 v_5) = 0$, $\kappa\ ({}^2 v_5) = 139$, ${}^1 v_5, {}^2 v_5 \in V(G)$, $\varepsilon\ (e_5) = 259$, $e_5 \in E(G_0)$

(iii) $\varepsilon\ ([{}^1 v_4,\ {}^1 v_{17}]) = \varepsilon\ ([{}^2 v_4,\ {}^2 v_{19}]) = 280$, ${}^1 v_4, {}^2 v_4$, ${}^1 v_{17}$, ${}^2 v_{19} \in V(G)$, $[{}^1 v_4, {}^1 v_{17}]$, $[{}^2 v_4, {}^2 v_{19}] \in E_{\text{trans}}\ (G)$

(iv) $\forall\ [{}^i v_x,\ {}^j v_y] \in E_{\text{rev}}\ (G)$: $\varepsilon\ ([{}^i v_x,\ {}^j v_y]) = 120$

The topology of the train route found (which corresponds to the shortest route in the digraph G and can be retransformed into the shortest walk in the graph G_0) is as follows:

$$e_4(350) \leftarrow e_{19}(316) \leftarrow e_7(276) \leftarrow e_{21}(238) \leftarrow e_8(118) \leftarrow e_{21}(80) \leftarrow e_7(40) \leftarrow e_{20}(0) \leftarrow e_5(0)$$

The numbers in parentheses (following each member of the walk) are the metric lengths of the relocation object's

```
function Shortest_Path(↓S_V, ↓F_V, ↓L, ↑Topol)
  Topol ← ∅
  correct ← true
  Start_Finish_Test(↓S_V, ↓F_V, ↓L, ↑↓correct) // admissibility of the start/finish vertices
  if correct then
    General_Init() // setting initial marks of all vertices
    Start_Finish_Init(↓S_V, ↓F_V) // initialisation activities related to the start/finish vertices
    Start_Mark(↓S_V, ↓L) // remarking of transit successors of the start/finish vertices
    repeat
      if T_V ≠ ∅ then
        Vert_Select(↑^k v_z) // selection of a new current vertex
        if U_V ≠ F_V then
          Succ_Mark(↓^k v_z, ↓L) // remarking successors of the current vertex ^k v_z
        end
      end
    until (T_V = ∅ or U_V = F_V) // algorithm termination testing
    Get_Path(↑↓Topol) // getting a topology of the shortest path
  end
end
function Start_Finish_Test(↓S_V, ↓F_V, ↓L, ↓↑okay)
  if (S_V = ∅ or F_V = ∅) then
    okay ← false
    exit
  end
  for each ^i v_x ∈ S_V do
    Get_Indexes(↓^i v_x, ↑i, ↑x)
    for each ^j v_y ∈ F_V do
      Get_Indexes(↓^j v_y, ↑j, ↑y)
      if (x = y and i ≠ j) then
        okay ← false // inadmissible combination of the start and finish vertices
        exit
      end
      if (ω (^i v_x) < L or κ (^j v_y) < L) then
        okay ← false // inadmissible weights/vacancies of the start/finish vertices
        exit
      end
    end
  end
end
function General_Init()
  for z = 1 to n\2 do // symbol "\" denotes an integer division
    for k = 1 to 2 do
      ^k d_z ← d^∞ // initialisation of the row vector of distance marks
      ^k p_z ← none // initialisation of the row vector of marks-predecessors
    end
  end
  T_V ← ∅
  U_V ← ∅
end
function Start_Finish_Init(↓S_V, ↓F_V)
  for each ^i v_x ∈ S_V do
    Get_Indexes(↓^i v_x, ↑i, ↑x)
    for each ^j v_y ∈ F_V do
      Get_Indexes(↓^j v_y, ↑j, ↑y)
      a ← (3 − i)
      b ← (3 − j)
      if S_V = F_V then
        X_V ← {^a v_x} // the forbidden vertex ^a v_x is a pair vertex to the start vertex ^i v_x
      else
        X_V ← {^a v_x, ^b v_y} // the forbidden vertex ^b v_y is a pair vertex to the finish vertex ^j v_y
        ^i d_x ← 0 // initialisation of selected distance marks
```

Algorithm 1: Continued.

```
        end
      end
    end
  end
  function Start_Mark(↓S_V, ↓L)
    for each ^k v_z ∈ S_V do
      for each ^l v_t ∈ ^out V_trans (^k v_z) do
        if (^l v_t ∉ X_V and κ (^l v_t) ≥ L) then
          T_V ← T_V ∪ {^l v_t} // insertion of admissible transit successors into the set T_V
          ^l d_t ← 0 // initialisation of selected distance marks
          ^l p_t ← ^k v_z // initialisation of selected marks-predecessors
        end
      end
    end
  end
  function Vert_Select(↑^k v_z)
    if T_V ≠ ∅ then
      Min_Dist(↓T_V, ↑^k v_z) // selection of a vertex ^k v_z with the lowest distance mark ^k d_z
      T_V ← T_V − {^k v_z}
      if ^k v_z ∈ F_V then
        U_V ← U_V ∪ {^k v_z}
      end
    end
  end
  function Try_Change_Mark(↓^k v_z, ↓^l v_s)
    Get_Indexes(↓^k v_z, ↑k, ↑z)
    Get_Indexes(↓^l v_s, ↑l, ↑s)
    if ^l d_s > ^k d_z + ε ([^k v_z, ^l v_s]) then
      ^l d_s ← ^k d_z + ε ([^k v_z, ^l v_s]) // remarking of the successor (^l v_s) of the vertex ^k v_z
      ^l p_s ← ^k v_z
      if ^l v_s ∉ T_V then
        T_V ← T_V ∪ {^l v_s}
      end
    end
  end
  function Succ_Mark(↓^k v_z, ↓L)
    for each ^l v_t ∈ ^out V_trans (^k v_z)do
      if (^l v_t ∉ X_V and κ (^l v_t) ≥ L and κ (^k v_z) = ω (^k v_z)) then
        Try_Change_Mark(↓^k v_z, ↓^l v_t) // potential remarking of the transit successor of ^k v_z
      end
    end
    for each ^l v_r ∈ ^out V_rev(^k v_z) do
      if (^l v_r ∉ X_V and κ (^l v_r) ≥ L) then
        Try_Change_Mark(↓^k v_z, ↓^l v_r) // potential remarking of the reverse successor of ^k v_z
      end
    end
  end
  function Get_Path(↑↓Seq)
    if U_V ≠ ∅ then
      Min_Dist(↓U_V, ↑^j v_y)
      Get_Indexes(↓^j v_y, ↑j, ↑y)
      z ← y
      k ← j
      i ← 0
      while (^k p_z ≠ none or ^k v_z ∈ S_V) do
        Seq ← Seq ∪ {[i, ^k v_z]} // successive reconstruction of the shortest path topology
        if ^k v_z ∈ S_V then
          exit
        end
        Predecessor(↓^k v_z, ↑l, ↑t) // getting a predecessor of the vertex ^k v_z
        z ← t
```

Algorithm 1: Continued.

```
        k ← l
        i ← i + 1
      end
    end
  end
```

Algorithm 1: Computation of the shortest path from $S_V = \{{}^i v_x\}$ to $F_V = \{{}^j v_y\}$

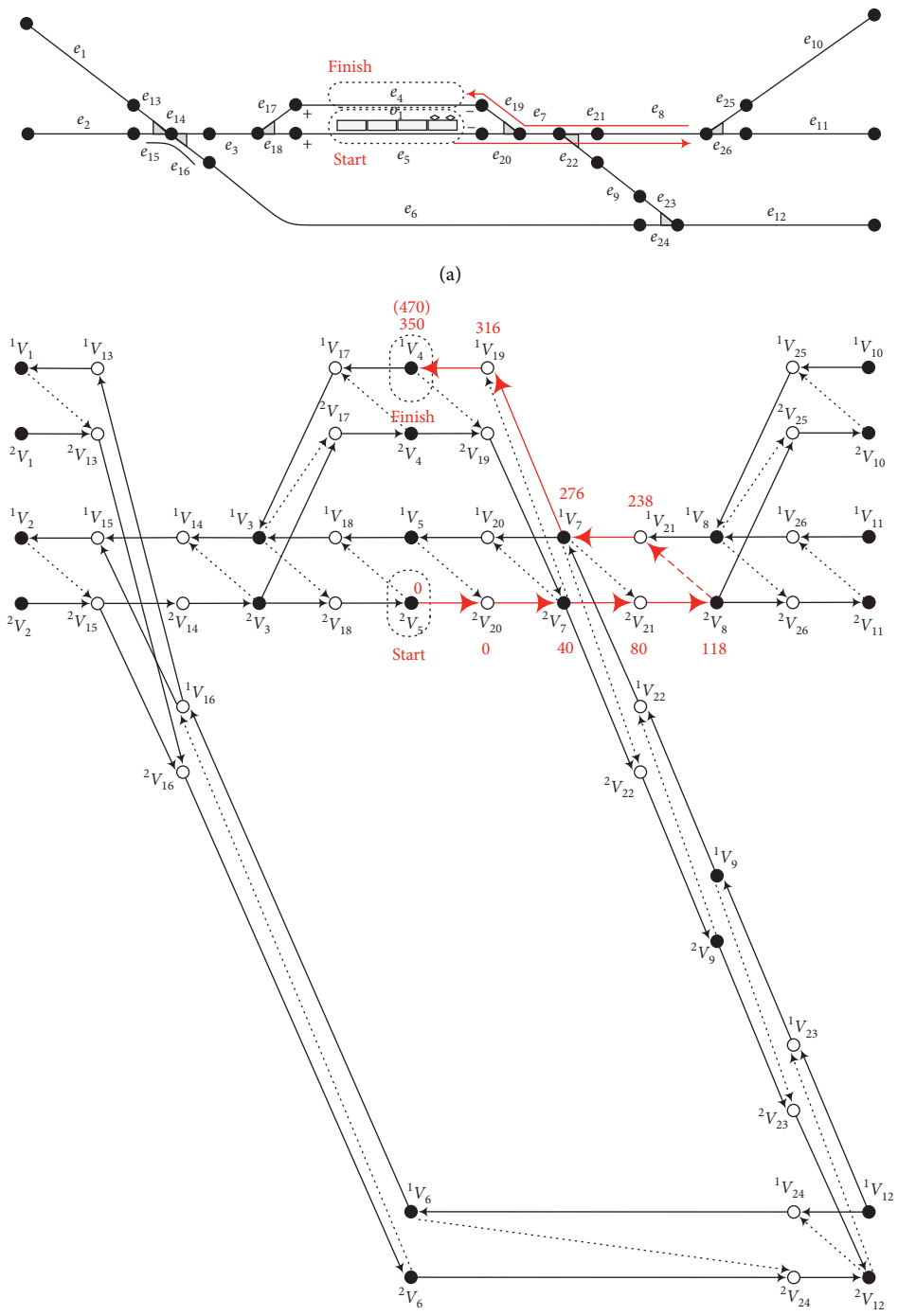
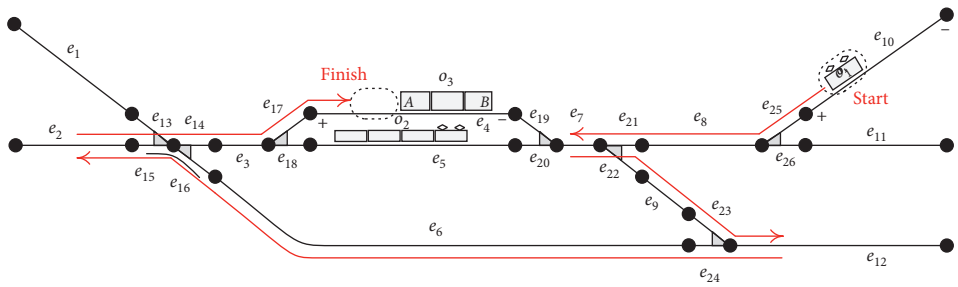
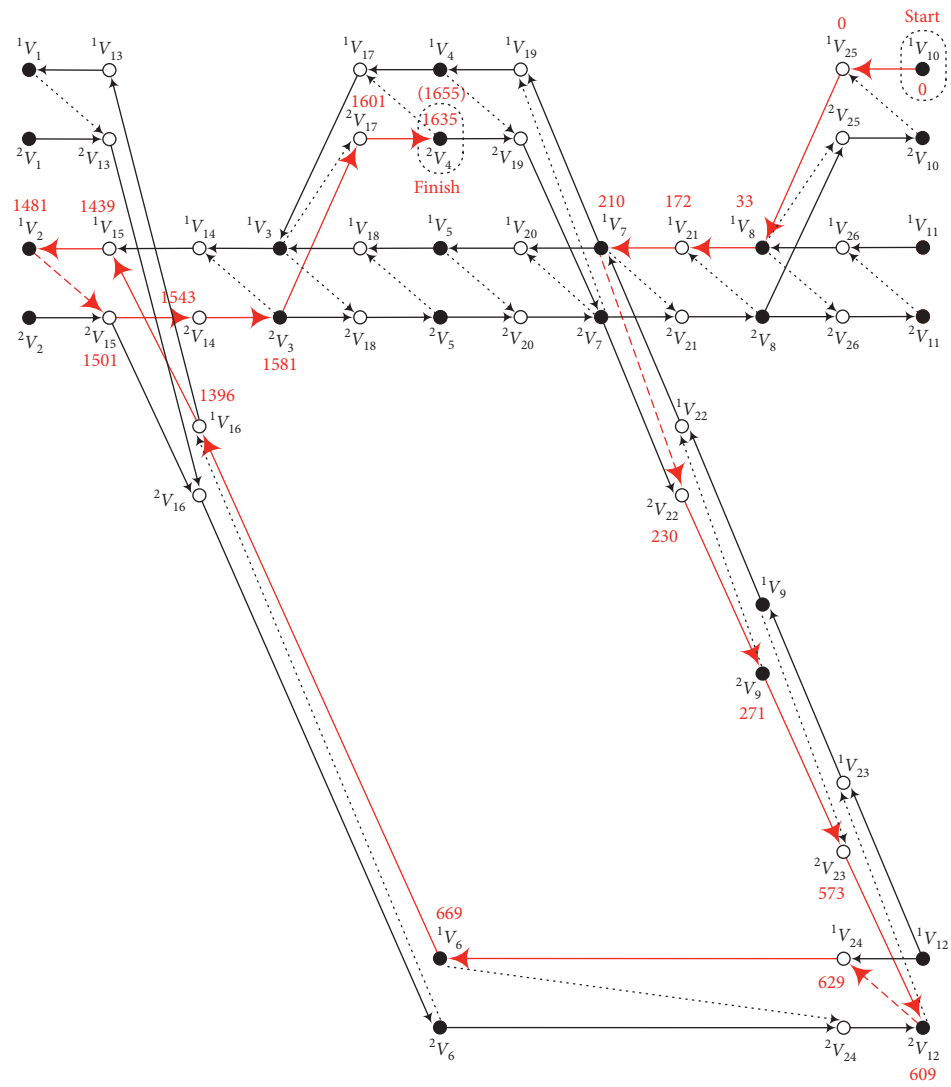


Figure 4: Train route related to Example 1.



(a)



(b)

Figure 5: Train route related to Example 2.

trajectory to the points representing the input ends of the edges/tracks reached. The total length of the train's trajectory during the relocation is 470 m. The value for the last member of the walk is additionally increased by $L = 120$ due to the fact that the relocation operation must be finished by a partial transfer of the whole object to the finish track (edge). For the computed shortest route in the digraph G, the figure shows the ultimate values of the distance marks for the relevant vertices.

Moreover, the reduced distance matrix (RDM) (for $L = 120$) between all the admissible couples of the destination vertices in the digraph G is shown in Table 7 (beyond the scope of Example 1). The RDM contains the results of differently parameterized computations made by using Algorithm 1. The matrix reflects the fact that, apart from the relocation object, which is always present on a different start track, the railway yard is entirely empty. The RDM is

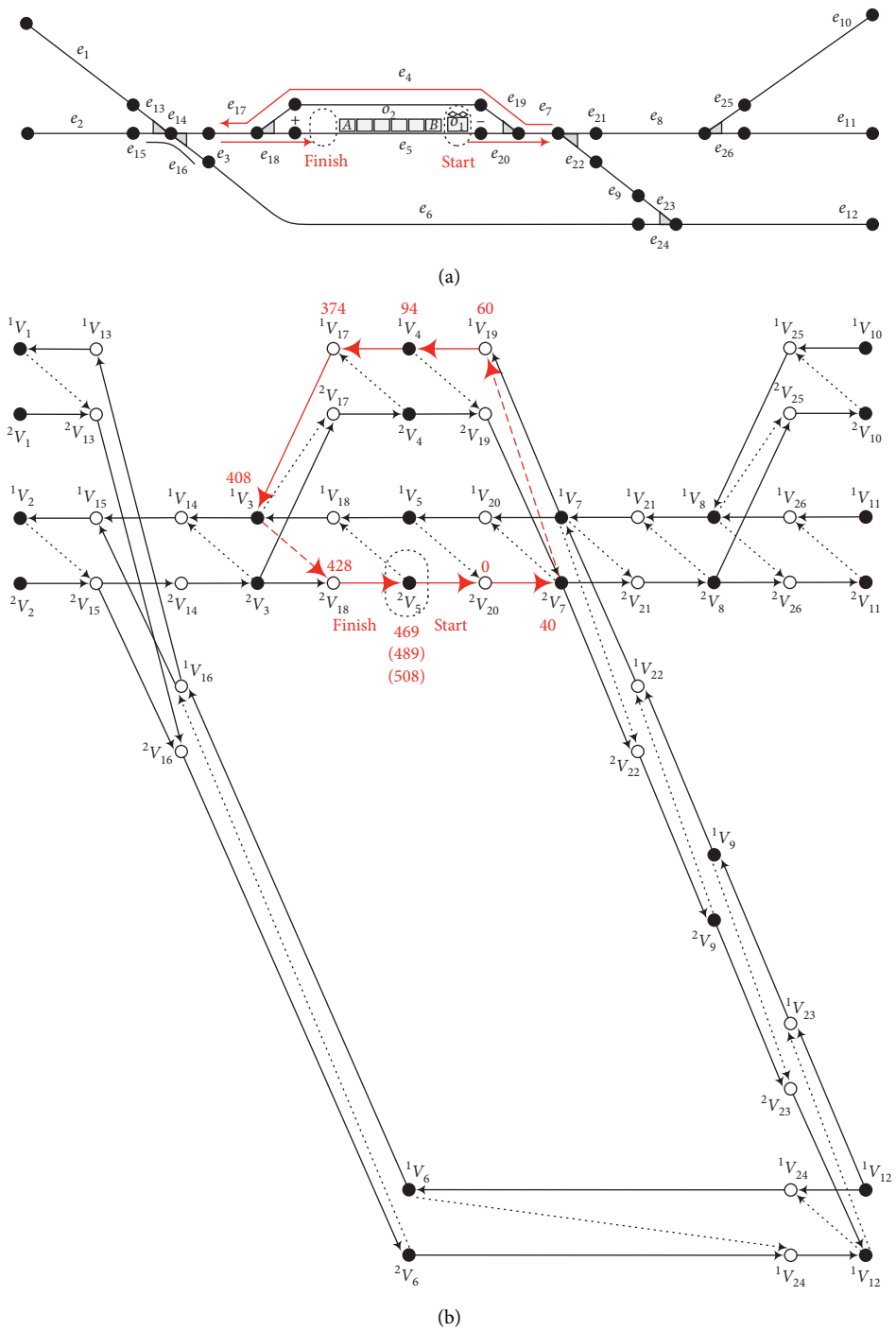


Figure 6: Train route related to Example 3.

obtained by simple transformation of the complete distance matrix (CDM). This transformation removes from the CDM those rows and columns whose all elements attain only the d^{∞} value (i.e., no shortest routes for the given parameter L exist between relevant vertices).

The next example concerns the motion of a shunting engine along a rather complex route for attachment to a specified end of a train set.

Example 2. Computation of a shunting route for the locomotive relocation ($L = 20$).

Figure 5 shows diagrams of graphs G_0 and G, which represent infrastructure models analogous to those in Figure 4. A 120 m-long train (object O_2) stands on track #5 of the yard; a 100 m-long train set (object O_3) stands on track #4; and a 20 m-long shunting locomotive (object O_1) stands on track #10. This locomotive is to be moved to end A of

Table 7: Reduced distance matrix (RDM) for the parameter $L = 120$ and vacant track layout.

From	To											
	${}^{-}e_1$ ${}^{1}v_1$	${}^{-}e_4$ ${}^{1}v_4$	${}^{-}e_5$ ${}^{1}v_5$	${}^{-}e_6$ ${}^{1}v_6$	${}^{+}e_6$ ${}^{2}v_6$	${}^{-}e_8$ ${}^{1}v_8$	${}^{+}e_8$ ${}^{2}v_8$	${}^{-}e_9$ ${}^{1}v_9$	${}^{+}e_9$ ${}^{2}v_9$	${}^{+}e_{10}$ ${}^{2}v_{10}$	${}^{+}e_{11}$ ${}^{2}v_{11}$	${}^{+}e_{12}$ ${}^{2}v_{12}$
${}^{-}e_1$ ${}^{2}v_1$	—	1533	1539	—	193	—	1765	1116	—	1937	1964	960
${}^{-}e_4$ ${}^{2}v_4$	1533	—	470	733	—	—	232	—	235	404	431	573
${}^{-}e_5$ ${}^{2}v_5$	1539	470	—	739	—	—	238	—	241	410	437	579
${}^{+}e_6$ ${}^{1}v_6$	193	—	—	—	—	—	—	—	—	—	—	—
${}^{-}e_6$ ${}^{2}v_6$	—	733	739	—	—	—	965	316	—	1137	1164	160
${}^{+}e_8$ ${}^{1}v_8$	1765	232	238	965	—	—	—	—	467	—	—	805
${}^{-}e_8$ ${}^{2}v_8$	—	—	—	—	—	—	—	—	—	153	180	—
${}^{+}e_9$ ${}^{1}v_9$	—	235	241	—	—	—	467	—	—	639	666	—
${}^{-}e_9$ ${}^{2}v_9$	1116	—	—	316	—	—	—	—	—	—	—	156
${}^{+}e_{10}$ ${}^{1}v_{10}$	1937	404	410	1137	—	153	—	—	639	—	333	977
${}^{+}e_{11}$ ${}^{1}v_{11}$	1964	431	437	1164	—	180	—	—	666	333	—	1004
${}^{+}e_{12}$ ${}^{1}v_{12}$	960	573	579	160	—	—	805	156	—	977	1004	—

train set O_3; in other words, it must enter track #4 by transiting its end ${}^{+}e_4$.

The track route computation algorithm (identifying the shortest route in the digraph G) is parameterized as follows:

(i) $S_V = \{{}^{1}v_{10}\}$, $F_V = \{{}^{2}v_4\}$, and $L = 20$

The weights are given for selected vertices and edges in the graphs, and current vacancies are quantified for selected vertices in the digraph G:

(i) $\omega\ ({}^{1}v_4) = \omega\ ({}^{2}v_4) = 280$, $\kappa\ ({}^{1}v_4) = 160$, $\kappa\ ({}^{2}v_4) = 20$, ${}^{1}v_4$, ${}^{2}v_4 \in V(G)$, $\varepsilon\ (e_4) = 280$, $e_4 \in E(G_0)$

(ii) $\omega\ ({}^{1}v_5) = \omega\ ({}^{2}v_5) = 259$, $\kappa\ ({}^{1}v_5) = 139$, $\kappa\ ({}^{2}v_5) = 0$, ${}^{1}v_5, {}^{2}v_5 \in V(G)$, $\varepsilon\ (e_5) = 259$, $e_5 \in E(G_0)$

(iii) $\omega\ ({}^{1}v_{10}) = \omega\ ({}^{2}v_{10}) = 175$, $\kappa\ ({}^{1}v_{10}) = 155$, $\kappa\ ({}^{2}v_{10}) = 0$, ${}^{1}v_{10}$, ${}^{2}v_{10} \in V(G)$, $\varepsilon\ (e_{10}) = 175$, $e_{10} \in E(G_0)$

(iv) $\forall\ [{}^{i}v_x, {}^{j}v_y] \in E_{\text{rev}}\ (G)$: $\varepsilon\ ([{}^{i}v_x, {}^{j}v_y]) = 20$

The topology of the train route found is as follows:

$e_4(1635) \leftarrow e_{17}(1601) \leftarrow e_3(1581) \leftarrow e_{14}(1543) \leftarrow e_{15}(1501) \leftarrow e_2(1481) \leftarrow e_{15}(1439) \leftarrow e_{16}(1396) \leftarrow e_6(669) \leftarrow e_{24}(629) \leftarrow e_{12}(609) \leftarrow e_{23}(573) \leftarrow e_9(271) \leftarrow e_{22}(230) \leftarrow e_7(210) \leftarrow e_{21}(172) \leftarrow e_8(33) \leftarrow e_{25}(0) \leftarrow e_{10}(0)$

The total length of the locomotive's relocation trajectory is 1655 m. The value given for the last member of the walk is increased by $L = 20$.

In the last example, shown in the following, a shunting locomotive should be moved from one end of a train set to the other end of that train set.

Example 3. Computation of a shunting route for reapproaching the same train set ($L = 20$).

A group of wagons, 200 m total length (object O_2), stands on track #5 of the yard whose models are shown in Figure 6. A 20 m-long shunting locomotive (object O_1) was disconnected from wagon group end B in order to attach it to end A of that group because this is needed for the planned operations. This means that the locomotive should reapproach the wagon group by transiting the end ${}^{+}e_5$ of track #5.

The track route computation algorithm (identifying the shortest route in the digraph G) is parameterized as follows:

$S_V = \{{}^{2}v_5\}$, $F_V = \{{}^{2}v_5\}$, and $L = 20$

The weights are given for selected vertices and edges in the graphs, and current vacancies are quantified for selected vertices in the digraph G:

(i) $\omega\ ({}^{1}v_5) = \omega\ ({}^{2}v_5) = 259$, $\kappa\ ({}^{1}v_5) = 0$, $\kappa\ ({}^{2}v_5) = 39$, ${}^{1}v_5$, ${}^{2}v_5 \in V(G)$, $\varepsilon\ (e_5) = 259$, $e_5 \in E(G_0)$

(ii) $\omega\ ({}^{1}v_{10}) = \omega\ ({}^{2}v_{10}) = 175$, $\kappa\ ({}^{1}v_{10}) = 155$, $\kappa\ ({}^{2}v_{10}) = 0$, ${}^{1}v_{10}$, ${}^{2}v_{10} \in V(G)$, $\varepsilon\ (e_{10}) = 175$, $e_{10} \in E(G_0)$

(iii) $\forall\ [{}^{i}v_x, {}^{j}v_y] \in E_{\text{rev}}\ (G)$: $\varepsilon\ ([{}^{i}v_x, {}^{j}v_y]) = 20$

The topology of the train route found is as follows:

$e_5(469) \leftarrow e_{18}(428) \leftarrow e_3(408) \leftarrow e_{17}(374) \leftarrow e_4(94) \leftarrow e_{19}(60) \leftarrow e_7(40) \leftarrow e_{20}(0) \leftarrow e_5(0)$

The trajectory run by locomotive O_1 from the starting position (${}^{-}e_5$) to the opposite end (${}^{+}e_5$) of the same track (track #5) is 469 m long. If the complete length of the

locomotive enters track #5, then it has run a trajectory of 489 m long. And it requires another 19 m for attachment to train set O_2, whereby the total trajectory length is 508 m.

For demonstration reasons (beyond the scope of Example 3), the RDM (for $L = 20$) for the destination vertices in the digraph G, containing results of the calculations for differently parameterized Algorithm 1, is included in Table 8. This matrix matches the situation where, apart from the relocation object (which is always present on a different start track), the railway yard is entirely empty. From this matrix, one can read that the trajectory of relocation between track #10 (its end ${}^{+}e_{10}$) and track #4 (its end ${}^{+}e_4$) is 664 m long. The shortest track route found passes through track #5. The route for the relocation (between the same tracks) is different (and differently long) from that in Example 2 because track #5 was occupied in that example.

4.3. Modifications of the Basic Algorithm. Conditions for the starting and destination positions different from those associated with Algorithm 1 may be used in the parameterization of the dynamic search of routes within a railway yard. The algorithm uses a uniquely defined track end from which the sought-for-train/shunting route should start, and this also applies to the end of the finish track. From the point of view of the final mathematical model (digraph G), it is exactly determined by one start vertex and exactly one finish vertex for the t route sought (thus, the algorithm searches for a single-source single-destination shortest path). In other words, the relation $|S_V| = |F_V| = 1$ holds for the set of start vertices S_V and the set of finish vertices F_V.

However, if the whole track x must be specified as the starting element (hence, leaving it through either of its ends is permissible), then the appropriate parameter of the *Shortest_Path* function is defined as $S_V = \{{}^1v_x, {}^2v_x\}$. The members of the set S_V correspond to the opposite ends of the track x in the digraph G. Analogously, the set F_V can be constructed as $F_V = \{{}^1v_y, {}^2v_y\}$ if the entire track is regarded as the destination element and entering it via either end is permissible. So, three different modifications of Algorithm 1, for different variants of construction of the sets S_V and F_V, are feasible. A summary overview of the *Shortest_Path* function parameterization options is presented in Table 9.

Algorithms 2–4 differ from the basic Algorithm 1 only due to changes in two subroutines, *Start_Finish_Test* and *Start_Finish_Init*, while the remaining parts are identical. The differences in the subroutines include different procedures of testing the admissibility of combinations of the input parameter values and different ways of performing initialisation actions associated with the start and finish vertices.

The first alternative to the basic algorithm is Algorithm 2, which seeks the shortest admissible route between two two-member sets of vertices (S_V and F_V) in the digraph G (two-sources two-destinations shortest path). One run of this algorithm provides the required route (through the function *Get_Path*), and the start vertex (from the set S_V) and finish vertex (from the set F_V) of that route are uniquely identified. The vertices define the direction in which the start track is left and the direction from which the finish track is entered. The *Start_Finish_Init* function potentially eliminates (among other things) from the set of finish vertices that vertex whose vacancy is inadequate for accommodating the relocation object whose length is L.

The next modification of the basic algorithm is Algorithm 3, computing the routes from two start vertices to one finish vertex in the digraph G (two-sources single-destination shortest path). When the process is over, the shorter of the two potentially computed routes is identified. The vertex ${}^bv_y$, which is the pair vertex to the vertex ${}^jv_y \in F_V$, is included into the set of forbidden vertices through the *Start_Finish_Init* function.

The last modification of the basic algorithm is Algorithm 4, seeking the shorter of two routes proceeding from one start vertex to two different finish vertices in the digraph G (single-source two-destinations shortest path). That vertex whose vacancy is inadequate with respect to parameter L is potentially eliminated from the set of finish vertices by the *Start_Finish_Init* function. In addition, that function augments the set of forbidden vertices with the vertex ${}^av_x$—the pair vertex to the vertex ${}^iv_x \in S_V$.

4.4. Verification and Validation. The life cycle of simulation studies/projects consists of a number of partial phases, as described in detail in [32, 33]. Discussed in the following is a part of the life cycle (consisting of several sequentially linked phases), with focus on the construction, verification, and validation of models that are typically used in simulation studies.

(a) Phase of designing and forming a conceptual model
(b) Conceptual model validation phase
(c) Phase of designing and building up a computerized model
(d) Computerized model verification phase
(e) Operational computerized model validation phase

The models can be briefly characterized as follows:

(i) The conceptual model reflects (with an appropriate degree of abstraction) the object of investigation
(ii) The computerized model represents an implementation of the conceptual model on a computer

In view of the scope of this article, attention (with respect to verification and validation) will not be paid to the whole target rail-traffic simulator (as implemented within the MesoRail tool): instead, only a part of it will be targeted, viz. that part that mirrors the rail infrastructure of the object of investigation and functions determined for computing track routes along which the rail vehicles in question can move on the rail infrastructure.

The conceptual model, which reflects the rail infrastructure of the object of investigation (representing the railway system in question) and the functions calculating track routes, uses the following:

Table 8: The RDM for the parameter $L = 20$ and vacant track layout.

From	To																		
	${}^{-}e_1$ ${}^{1}v_1$	${}^{-}e_2$ ${}^{1}v_2$	${}^{-}e_3$ ${}^{1}v_3$	${}^{+}e_3$ ${}^{2}v_3$	${}^{-}e_4$ ${}^{1}v_4$	${}^{+}e_4$ ${}^{2}v_4$	${}^{-}e_5$ ${}^{1}v_5$	${}^{+}e_5$ ${}^{2}v_5$	${}^{-}e_6$ ${}^{1}v_6$	${}^{+}e_6$ ${}^{2}v_6$	${}^{-}e_7$ ${}^{1}v_7$	${}^{+}e_7$ ${}^{2}v_7$	${}^{-}e_8$ ${}^{1}v_8$	${}^{+}e_8$ ${}^{2}v_8$	${}^{-}e_9$ ${}^{1}v_9$	${}^{+}e_9$ ${}^{2}v_9$	${}^{+}e_{10}$ ${}^{2}v_{10}$	${}^{+}e_{11}$ ${}^{2}v_{11}$	${}^{+}e_{12}$ ${}^{2}v_{12}$
${}^{-}e_1$ ${}^{2}v_1$	—	198	1639	298	712	352	726	359	—	93	1259	658	—	736	916	739	908	935	860
${}^{-}e_2$ ${}^{2}v_2$	198	—	1651	100	514	154	528	161	939	105	1271	460	—	538	928	541	710	737	872
${}^{+}e_3$ ${}^{1}v_3$	298	100	1751	—	1445	—	1451	—	—	205	1371	—	—	1429	1028	—	1601	1628	972
${}^{-}e_3$ ${}^{2}v_3$	1639	1651	—	1751	414	54	428	61	839	—	—	360	—	438	—	441	610	637	779
${}^{+}e_4$ ${}^{1}v_4$	352	154	54	—	468	—	1505	115	893	259	1425	414	—	492	1082	495	664	691	833
${}^{-}e_4$ ${}^{2}v_4$	712	514	414	1445	—	468	114	1506	533	619	—	54	—	132	1442	135	304	331	473
${}^{+}e_5$ ${}^{1}v_5$	359	161	61	—	1506	115	489	—	908	266	1432	429	—	507	1089	510	679	706	848
${}^{-}e_5$ ${}^{2}v_5$	726	528	428	1451	114	1505	—	489	539	633	—	60	—	138	1456	141	310	337	479
${}^{+}e_6$ ${}^{1}v_6$	93	105	—	205	619	259	633	266	1044	—	—	565	—	643	—	646	815	842	984
${}^{-}e_6$ ${}^{2}v_6$	—	939	839	—	533	893	539	908	—	1044	459	—	—	517	116	—	689	716	60
${}^{+}e_7$ ${}^{1}v_7$	658	460	360	—	54	414	60	429	—	565	1731	—	—	—	1388	—	—	—	1332
${}^{-}e_7$ ${}^{2}v_7$	1259	1271	—	1371	—	1425	—	1432	459	—	—	1731	—	58	—	61	230	257	399
${}^{+}e_8$ ${}^{1}v_8$	736	538	438	1429	132	492	138	507	517	643	58	—	—	—	1466	119	—	—	457
${}^{-}e_8$ ${}^{2}v_8$	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	53	80	—
${}^{+}e_9$ ${}^{1}v_9$	739	541	441	—	135	495	141	510	—	646	61	—	—	119	1469	—	291	318	1413
${}^{-}e_9$ ${}^{2}v_9$	916	928	—	1028	1442	1082	1456	1089	116	—	—	1388	—	1466	—	1469	1638	1665	56
${}^{+}e_{10}$ ${}^{1}v_{10}$	908	710	610	1601	304	664	310	679	689	815	230	—	53	—	1638	291	—	133	629
${}^{+}e_{11}$ ${}^{1}v_{11}$	935	737	637	1628	331	691	337	706	716	842	257	—	80	—	1665	318	133	—	656
${}^{+}e_{12}$ ${}^{1}v_{12}$	860	872	779	972	473	833	479	848	60	984	399	1332	—	457	56	1413	629	656	—

Table 9: Variants of algorithms for calculating the shortest paths.

	Start vertices $S_V \subset V(G)$	Finish vertices $F_V \subset V(G)$	Search type
Algorithm 1	$S_V = \{{}^{i}v_x\}$	$F_V = \{{}^{j}v_y\}$	Single-source single-destination
Algorithm 2	$S_V = \{{}^{1}v_x, {}^{2}v_x\}$	$F_V = \{{}^{1}v_y, {}^{2}v_y\}$	Two-sources two-destinations
Algorithm 3	$S_V = \{{}^{1}v_x, {}^{2}v_x\}$	$F_V = \{{}^{j}v_y\}$	Two-sources single-destination
Algorithm 4	$S_V = \{{}^{i}v_x\}$	$F_V = \{{}^{1}v_y, {}^{2}v_y\}$	Single-source two-destinations

(i) A mathematical model weighted digraph, as described in the Final Model section. The digraph is a result of transformation of the primary mathematical model—a weighted undirected graph. The primary model reflects rather intuitively the topological and metric situation of the rail infrastructure associated with the object of investigation.

(ii) The concept of Dijkstra's algorithm for searching for the shortest paths in graphs. The algorithm had to be modified appreciably for use in problems of determining track routes on the final infrastructure model.

Validation of the conceptual model included, in particular, assessment of suitability of both mathematical models for description of the relevant part of the object of investigation and of the algorithms reflecting the selected operations on the object of investigation. This phase used the IV&V (independent verification and validation) approach [33], applied after completing the development of the

```
function Start_Finish_Test(↓S_V, ↓F_V, ↓L, ↓↑okay)
  if (S_V = ∅ or F_V = ∅ or S_V = F_V) then
    okay ← false
    exit
  end
  for each ^i v_x ∈ S_V do
    if (ω (^i v_x) < L or (κ (^1 v_y) < L and κ (^2 v_y) < L)) then
      okay ← false
      exit
    end
  end
end
function Start_Finish_Init(↓S_V, ↓F_V)
  for each ^j v_y ∈ F_V do
    if κ (^j v_y) < L then
      F_V = F_V − {^j v_y} // exclusions of finish vertices with insufficient vacant capacities
    end
  end
  for each ^i v_x ∈ S_V do
    Get_Indexes(↓^i v_x, ↑i, ↑x)
    ^i d_x ← 0
  end
  X_V ← ∅
end
```

Algorithm 2: Modification of Algorithm 1 for $S_V = \{^1v_x, {}^2v_x\}$ and $F_V = \{^1v_y, {}^2v_y\}$.

```
function Start_Finish_Test(↓S_V, ↓F_V, ↓L, ↓↑okay)
  if (S_V = ∅ or F_V = ∅ or F_V ⊂ S_V) then
    okay ← false
    exit
  end
  for each ^i v_x ∈ S_V do
    for each ^j v_y ∈ F_V do
      if (ω (^i v_x) < L or κ (^j v_y) < L) then
        okay ← false
        exit
      end
    end
  end
end
function Start_Finish_Init(↓S_V, ↓F_V)
  for each ^i v_x ∈ S_V do
    Get_Indexes(↓^i v_x, ↑i, ↑x)
    ^i d_x ← 0
  end
  Get_Indexes(↓^j v_y, ↑j, ↑y)
  b ← (3 − j)
  X_V ← {^b v_y} // a forbidden vertex ^b v_y represents a pair vertex to the finish vertex ^j v_y
end
```

Algorithm 3: Modification of Algorithm 1 for $S_V = \{^1v_x, {}^2v_x\}$ and $F_V = \{^jv_y\}$.

conceptual model. This implied in practice that the validation was made by an independent third-party professional (from the Czech Railway Infrastructure Administration, State Organization), who was a renowned expert in the railway traffic domain and in the application of computer simulations for the needs of railway traffic optimizations (and was the second author of this paper). The face validation (expert validation) [33] method was used, requiring

```
function Start_Finish_Test(↓S_V, ↓F_V, ↓L, ↓↑okay)
    if (S_V = ∅ or F_V = ∅ or S_V ⊂ F_V) then
        okay ← false
        exit
    end
    for each ^i v_x ∈ S_V do
        if (ω (^i v_x) < L or (κ (^1 v_y) < L and κ (^2 v_y) < L)) then
            okay ← false
            exit
        end
    end
end
function Start_Finish_Init(↓S_V, ↓F_V)
    for each ^j v_y ∈ F_V do
        if κ (^j v_y) < L then
            F_V = F_V − {^j v_y} // exclusions of finish vertices with insufficient vacant capacities
        end
    end
    Get_Indexes(↓^i v_x, ↑i, ↑x)
    ^i d_x ← 0
    a ← (3 − i)
    X_V ← {^a v_x} // a forbidden vertex ^a v_x represents a pair vertex to the start vertex ^i v_x
end
```

Algorithm 4: Modification of Algorithm 1 for $S_V = \{^i v_x\}$ and $F_V = \{^1 v_y, {}^2 v_y\}$.

Table 10: Mean calculation times of Algorithm 1 related to searching a shortest path.

	$\lvert V(G)\rvert$ (—)	$\lvert E(G)\rvert$ (—)	L (—)	Calculated paths (—)	^{mean}t (ms)
Demonstration railway yard Total length of tracks: 3.05 km	52	78	0	2,652	0.02
			20	2,652	0.02
			50	462	0.03
			100	380	0.03
			200	56	< 0.01
			300	12	< 0.01
Real railway yard Total length of tracks: 75.96 km	1178	1905	0	1,386,506	5.40
			20	184,470	5.30
			50	71,556	5.00
			100	38,220	4.80
			200	23,562	4.80
			300	17,292	4.80
			400	12,882	4.70
			500	11,990	4.60

that the professional performing the validation has deep knowledge of the relevant application domain (railway traffic in this case). The conclusions from the conceptual model validation process were as follows: (i) the designed mathematical models of the rail infrastructure adequately reflect the characteristics of the real railway infrastructure; and (ii) the algorithms for track route computations relating to rail vehicle shunting and train runs have been correctly logically designed so as to be able to provide outputs (i.e., specific track routes) usable in actual railway systems.

When developing the computerized model, data structures (mentioned in the conclusion of the Basic Algorithm section) were selected so as to be applicable to the implementation of the weighted digraph (reflecting the rail infrastructure) and to enable effective computations of the implemented track route searching algorithms.

The computerized model verification phase included testing of the model's logical correctness. This means that the track routes found for the specific objects of relocation were tested especially with respect to the following conditions: (i) the length of the relocation object is respected; (ii) those infrastructure elements that are currently occupied by rail vehicles or blocked by the interlocking system are not used; and (iii) reversals are performed on admissible track segments. The testing was made on different data instances (describing 3 different rail yards) applying different relocation object lengths and differently occupied track layouts. Verification was made by the person who had developed the

computerized model (and is the first author of this paper). The performance of the algorithms was found to be logically correct (after eliminating a few minor errors).

The IV&V approach and the face validation method were also applied (by the same expert as during the conceptual model validation phase) during the operational validation of the computerized model. The expert assessment of results of the track route computations (using the same track layout variant as in the computerized model verification phase) included, in particular, assessment of the operational and technical usability of the track routes identified (for different relocation object lengths) for the track layouts as currently occupied. Following analysis of all the cases of track routes examined (as obtained by using the original algorithms for the computations of the shortest paths on the weighted digraph), the expert concluded that the algorithms performed correctly from both operational and technical aspects. Based on that, this partial computerized model was embedded in the target simulator within the MesoRail software tool.

4.5. Technical Notes for Practical Use of Algorithms within Rail-Traffic Simulators. A few technical comments should be added regarding practical use of the algorithms within software tools serving to simulate railway traffic (at the mesoscopic level of detail):

(a) A combined approach can be applied in the simulators when searching for the optimal train routes and shunting routes. This may include selecting from a precalculated static set of essential train routes (constructed before starting the simulation experiments) and application of dynamic computation during the simulation when seeking for the optimum shunting routes. In this manner, the simulation will not be burdened by dynamic search for selected (very frequently repeating) train routes.

(b) It is typical of railway traffic that dedicated track routes are not set on the infrastructure between extremely distant start and finish tracks. Instead, the approach of stepwise construction of several shorter routes is preferred. Taking this into account, the algorithms discussed can be augmented with a limitation regarding the maximum admissible length of the track route being sought. This limitation can be used to put restrictions to the spread of the computation within the appropriate graph. This approach can eliminate undesirable instances of very complex and long shunting routes (found, e.g., in a densely occupied part of the track infrastructure). This means in practice that the algorithm may include identification of the lowest value ($d^{\min}$) among all distance marks for the vertices currently included in the set T_V. The $d^{\min}$ value can be indicated simply when withdrawing a vertex from the set T_V through the function *Vert_Select.* If this value exceeds an expertly defined limiting value $d^{\max}$, the computation is terminated with the result that no route was found. For the relocation object, this means that, given the current traffic situation, it will have to stay where it is for some time until the traffic situation changes (e.g., when the relevant tracks become available for the operation), and the search for the track route can be repeated.

(c) Given the degree of abstraction applied, the algorithms presented work with a simplification against reality, mainly concerning reversals within the motion of a relocation object. A reversal is defined as a situation where the relocation object stops at a certain position and then continues its motion in the direction opposite to the initial direction. As mentioned above, the infrastructure model presented and the appropriate algorithms always consider a reversal as an operation occurring on one track. In reality, however, more than one track (and switch) can be engaged in the reversal of a relocation object. The simplification is appropriate for simulators working at the mesoscopic level of detail (intended, e.g., for examining the capacity of the infrastructure) because shunting operations, in particular, are not supposed to be examined in great detail. However, when examining railway traffic at the microscopic level, such simplification might provide an unsatisfactory solution, unusable for application.

(d) For practical use of the algorithms in simulation experiments, one must have an idea of how much real time the computations take—such information is important for the assessment of their usability in the simulations because they should not be very time consuming. By way of illustration of the time demands, computation experiments were performed for Algorithm 1 (with the highest potential for use in traffic simulations) in a model of a basic railway yard (shown in Figure 3) and in a more extensive model of a larger real railway marshalling yard in Teplička nad Váhom, Slovakia. The latter model (represented by a digraph) contained 1178 vertices (reflecting 589 track segments) and 1905 edges (the total length of tracks in the infrastructure was 75,958 m). Different values of parameter L (reflecting the relocation object lengths) were applied, viz. $L = 0$, 20, 50, 100, 200, 300, 400, and 500. The shortest routes were computed for all admissible vertex couples, which satisfied the condition that their weight (or vacancy) was not lower than the applied value of parameter L (both models reflected empty railway yards). This, however, also means that—particularly for the more extensive model—such complex routes were computed as would never be actually computed within the simulators. Such routes were included in the experiments for testing purposes only. And in addition, connecting vertices were also considered as the start vertices and finish vertices (except for destination vertices) in order to increase the number of the routes computed.

The results of the calculations providing mean durations ($^{\text{mean}}t$) of searches for the shortest route are listed in Table 10. Generally, the mean time of computation of one shortest route within a more extensive demonstration model

does not exceed 6 milliseconds, which is a very good figure with respect to the needs of railway traffic simulators. The computations were made on a common PC equipped with an Intel i7-8550U CPU@1.80 GHz processor and 16.0 GB RAM.

The results also reflect computations that did not find the shortest route because none exists. The search for the shortest route between the vertices ${}^{2}v_5$ and ${}^{1}v_{10}$ for $L = 100$ in the model shown in Figure 3 is an example: although both vertices have a sufficient vacancy, the vertex ${}^{1}v_{10}$ is inaccessible because it mirrors that end of track #10 that constitutes the boundary of the model.

5. Conclusions

In conclusion, it has been demonstrated that the presented algorithms for automated dynamic calculations of the shortest track route topologies can be used with advantage within mesoscopic rail-traffic simulators to solve current traffic problems. The routes can then be used for shunting postulated relocation objects within the railway infrastructure in the actual occupancy situation. The data to be input when searching for the route include the start and finish positions and length of the object to be relocated.

So, the algorithms presented offer a good potential for extending and improving the scope of mesoscopic railway traffic simulators as regards automated identification of track routes especially for rail vehicle shunting.

The data specification of the rail infrastructure configuration, which currently uses the XML format, can be transformed in the future into a data description that will be fully compliant with the railML standard.

From the managerial aspect, it is noteworthy that it is convenient for the simulation projects in the railway traffic domain to acquire and use such simulation tools as support rapid constructions of the simulation models and fast parameterizations of the simulation experiments. Hence, if the simulation tool selected is equipped with functionalities that automatically compute track routes for specific relocation objects during the run of the simulation program, then such a tool has a comparative advantage over other tools lacking such functionalities. This is so because in this case, the user is freed from the requirement to tediously predefine many track routes, and hence, the time of the development of the target simulation models reflecting the operation of the railway system examined is appreciably shortened.

Data Availability

There is no supplementary information attached to the research article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research presented in this article was supported by the ERDF/ESF Project: Cooperation in Applied Research between the University of Pardubice and companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans) (no. CZ.02.1.01/0.0/0.0/17_049/0008394).

References

[1] J.-F. Cordeau, P. Toth, and D. Vigo, "A survey of optimization models for train routing and scheduling," *Transportation Science*, vol. 32, no. 4, pp. 380–404, 1998.

[2] A. Almech, E. Roanes-Lozano, C. Solano-Macías, and A. Hernando, "A new approach to shortest route finding in a railway network with two track gauges and gauge change-overs," *Mathematical Problems in Engineering*, vol. 2019, Article ID 8146150, 16 pages, 2019.

[3] L. G. Kroon, H. Edwin Romeijn, and P. J. Zwaneveld, "Routing trains through railway stations: complexity issues," *European Journal of Operational Research*, vol. 98, no. 3, pp. 485–498, 1997.

[4] R. Freling, R. M. Lentink, L. G. Kroon, and D. Huisman, "Shunting of passenger train units in a railway station," *Transportation Science*, vol. 39, no. 2, pp. 261–272, 2005.

[5] J.-A. Adlbrecht, B. Hüttler, N. Ilo, and M. Gronalt, "Train routing in shunting yards using Answer Set Programming," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7292–7302, 2015.

[6] J. Riezebos and W. Van Wezel, "K-Shortest routing of trains on shunting yards," *OR Spectrum*, vol. 31, no. 4, pp. 745–758, 2009.

[7] A. Kavicka and L. Janosikova, "Trackage modelling and algorithms for finding the shortest train route," *Communications—Scientific Letters of the University of Zilina*, vol. 1, no. 2, pp. 9–21, 1999.

[8] M. Montigel, *Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen*, ETH Zurich, Zürich, Switzerland, 1994.

[9] B. Zelinka, "Polar graphs and railway traffic," *Applications of Mathematics*, vol. 19, no. 3, pp. 169–176, 1974.

[10] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[11] T. Cormen, *Introduction to Algorithms*, MIT Press, Cambridge, MA, USA, 2nd edition, 2001.

[12] M. Barbehenn, "A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices," *IEEE Transactions on Computers*, vol. 47, no. 2, 1998.

[13] F. Schulz, D. Wagner, and K. Weihe, "Dijkstra's algorithm online," *ACM Journal of Experimental Algorithmics*, vol. 5, p. 12, 2000.

[14] F. Schulz, D. Wagner, C. Zaroliagis, C. Stein, and D. M. Mount, "Using multi-level graphs for timetable information in railway systems," *Algorithm Engineering and Experiments (Lecture Notes in Computer Science)*, Springer, Berlin, Germany, pp. 7–12, 2002.

[15] D. Wang, X. Chen, and H. Huang, "A graph theory-based approach to route location in railway interlocking," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 791–799, 2013.

[16] G. Medeossi and S. De Fabris, "Simulation of rail operations," in *Handbook of Optimization in the Railway Industry*, R. Borndörfer, T. Klug, L. Lamorgese et al., Eds., Springer International Publishing, Cham, Switzerland, pp. 1–24, 2018.

[17] A. Nash and D. Huerlimann, "Railroad simulation using OpenTrack," *WIT Transactions on the Built Environment*, vol. 74, 2004.

[18] A. Radtke and J.-P. Bendfelt, *Handling of Railway Operation Problems with RailSys*, Institute of Transport, University of Hanover, Hanover, Germany, 2011.
[19] N. Adamko and V. Klima, "Optimisation of railway terminal design and operations using Villon generic simulation model," *Transport*, vol. 23, no. 4, pp. 335–340, 2008.
[20] R. Divis and A. Kavicka, "Design and development of a mesoscopic simulator specialized in investigating capacities of railway nodes," in *Proceedings of the 27th European Modeling and Simulation Symposium, EMSS 2015*, pp. 52–57, Bergeggi, Italy, September 2015.
[21] B. Sewcyk and M. Kettner, *Network Evaluation Model NEMO*, Institute of Transport. Germany: University of Hanover, Hanover, Germany, 2001.
[22] Y. Cui, U. Martin, and J. Liang, "PULSim: user-based adaptable simulation tool for railway planning and operations," *Journal of Advanced Transportation*, vol. 2018, Article ID 7284815, 11 pages, 2018.
[23] Y. Cui and U. Martin, "Multi-scale simulation in railway planning and operation," *PROMET—Traffic&Transportation*, vol. 23, no. 6, pp. 511–517, 2011.
[24] R. Novotny and K. Antonin, "Unitary hybrid model of railway traffic," in *Proceedings of the 29th European Modeling and Simulation Symposium, EMSS 2017, International Multidisciplinary Modeling and Simulation Multiconference, I3M 2017*, pp. 181–186, Barcelona, Spain, September 2017.
[25] W. Burghout, H. N. Koutsopoulos, and I. Andreasson, "A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems, ITSC 2006*, pp. 1102–1107, Maui, HI, USA, November 2006.
[26] X. Chen, S. He, T. Li, and Y. Li, "A simulation platform for combined rail/road transport in multiyards intermodal terminals," *Journal of Advanced Transportation*, vol. 2018, Article ID 5812939, 19 pages, 2018.
[27] https://www.railml.org/en/.
[28] T. Ciszewski and W. Nowakowski, "RailML as a tool for description of data model in railway traffic control systems," in *Proceedings of the International Conference-Transport Means*, pp. 935–940, Kaunas University of Technology, Trakai, Lithuania, October 2018.
[29] http://www.railtopomodel.org/en/.
[30] A. Hlubuček, "RailTopoModel and railML 3 in overall context," *Acta Polytechnica CTU Proceedings*, vol. 11, pp. 16–21, 2017.
[31] J. Ebert, "A versatile data structure for edge-oriented graph algorithms," *Communications of the ACM*, vol. 30, no. 6, pp. 513–519, 1987.
[32] J. Banks, *Handbook of Simulation [Online]*, John Wiley & Sons, Hoboken, NJ, USA, 1998.
[33] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings of the 2010 Winter Simulation Conference*, pp. 166–183, IEEE, Baltimore, MD, USA, December 2010.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 8819094, 11 pages
https://doi.org/10.1155/2021/8819094

Research Article

Identifying and Labeling Potentially Risky Driving: A Multistage Process Using Real-World Driving Data

Charles Marks,[1] **Arash Jahangiri**,[2] **and Sahar Ghanipoor Machiani**[2]

[1]*Interdisciplinary Research on Substance Use Joint Doctoral Program, San Diego State University and the University of California San Diego, San Diego, CA, USA*
[2]*Department of Civil, Construction, and Environmental Engineering, San Diego State University, San Diego, CA, USA*

Correspondence should be addressed to Arash Jahangiri; ajahangiri@sdsu.edu

Received 24 August 2020; Revised 15 December 2020; Accepted 27 January 2021; Published 15 February 2021

Academic Editor: Ladislav Routil

Every year, over 50 million people are injured and 1.35 million die in traffic accidents. Risky driving behaviors are responsible for over half of all fatal vehicle accidents. Identifying risky driving behaviors within real-world driving (RWD) datasets is a promising avenue to reduce the mortality burden associated with these unsafe behaviors, but numerous technical hurdles must be overcome to do so. Herein, we describe the implementation of a multistage process for classifying unlabeled RWD data as potentially risky or not. In the first stage, data are reformatted and reduced in preparation for classification. In the second stage, subsets of the reformatted data are labeled as potentially risky (or not) using the Iterative-DBSCAN method. In the third stage, the labeled subsets are then used to fit random forest (RF) classification models—RF models were chosen after they were found to be performing better than logistic regression and artificial neural network models. In the final stage, the RF models are used predictively to label the remaining RWD data as potentially risky (or not). The implementation of each stage is described and analyzed for the classification of RWD data from vehicles on public roads in Ann Arbor, Michigan. Overall, we identified 22.7 million observations of potentially risky driving out of 268.2 million observations. This study provides a novel approach for identifying potentially risky driving behaviors within RWD datasets. As such, this study represents an important step in the implementation of protocols designed to address and prevent the harms associated with risky driving.

1. Introduction

Each year, globally, traffic accidents result in 1.35 million deaths and 50 million injuries [1]. In 1998 in the United States, the National Highway Traffic Safety Administration (NHSTA) identified that aggressive driving behaviors occur in approximately two-thirds of all fatal car accidents [2]. Since then, multiple studies have corroborated the connection between aggressive driving behaviors and fatal car crashes [3–8]. The AAA Foundation found that, from 2003–2007, over half of fatal accidents were the result of aggressive driving behaviors [9]. In order to reduce the harms of aggressive driving behaviors, novel strategies for identifying such driving behaviors are required.

The concept of "aggressive driving" was formally defined in Meyer Parry's 1968 work, "Aggression on the Road," in which he stated that "the increasing stress involved in motoring nowadays makes the psychological efficiency of the driver a more important factor than the mechanical efficiency of the vehicle he drives" [10]. Looking at several studies on the topic, there is not a formal consensus on the definition of aggressive driving, but it ranges from acts of carelessness and recklessness to "road rage" [11–14]. One definition which captures these varying conceptions of aggressive driving is as follows: "A driving behavior is aggressive if it is deliberate, likely to increase the risk of collision, and is motivated by impatience, annoyance, hostility, and/or an attempt to save time" [15]. Since it is not usually possible to accurately assess the impatience,

annoyance, or attitude of drivers at scale, it is generally simpler to focus on the middle of this definition—driving behaviors which increase the risk of collision. Therefore, the term "risky driving" was used in the present study instead of "aggressive driving." However, since aggressive driving has been used in several previous studies, the same terminology was used when referring to those.

While examples of risky driving, such as tailgating, running red lights, and speeding, are easily recognized [15], in practice, identifying real-world risky driving at scale is complicated by a lack of both data and strategies to properly assess said data. A video may catch a car running a red light and a GPS unit may record that its vehicle is speeding, but the steps required to take available data and identify patterns of risky driving behaviors require innovative strategies. This is especially important when dealing with "big data," which is currently limited in the transportation research literature.

With advances in technologies, the ability to collect large quantities of real-world driving data (RWD, such as the speed, acceleration, and heading of a vehicle across entire trips) has greatly increased. The use of machine learning strategies to try to identify and classify aggressive driving behaviors within these large RWD datasets is a field of budding interest. An array of supervised learning methods such as linear regression [16, 17], naïve Bayes classification [18], support vector machines [19], artificial neural networks [19, 20], dynamic time warping with k-nearest neighbors [21], random forests [22], and deep learning approaches [23] has been used to classify RWD data as either aggressive or not. Unsupervised methods such as k-means [24, 25], self-organizing maps (a type of unsupervised neural network) [25], and DBSCAN [26] have been incorporated into aggressive driving classification efforts, as well.

These studies represent important advancements in the efforts to identify aggressive driving from RWD data. Feng et al. used the measurement of longitudinal jerk in order to identify aggressive driving behaviors [16]. Wang et al. created an index to identify jerky driving movements as potential indicators of aggressive driving [17]. Jahangiri et al. identified aggressive driving while negotiating turns by modeling vehicles crossing lane stripes [22]. Several studies used RWD data collected from smartphones [18, 19, 21, 27]. Hong et al. and Johnson et al. used RWD data from smartphones to identify aggressive driving styles [18, 21]. Yu et al. identified the statistical profiles of specific types of aggressive driving (e.g., weaving, slamming the breaks, etc.) and used smartphone RWD data to train models to identify these behaviors [19]. Jeihani et al. leveraged a series of machine learning strategies to identify observations characterized by sudden changes in statistical profiles (i.e., sudden drops in speed and sudden turns) [28].

While these endeavors represent important steps in mitigating the harms of risky driving, for agencies and organizations dedicated to improving traffic safety, these individual studies do not provide a full account of all the necessary steps (such as restructuring RWD data for analysis and accounting for the large size of RWD data via time- and memory-efficient algorithmic choices) to identify risky driving behaviors from RWD data. Providing a guide to the implementation of risky driving classification strategies is necessary to ensure that agencies are empowered to utilize such strategies to improve traffic safety within their jurisdictions.

The overall purpose of this study is to demonstrate a multistage process for classifying observations in a large RWD dataset as potentially risky or not, using kinematic data only. We present four distinct stages in which the process is divided: formatting the data for analysis; labeling a subset of the data as potentially risky or not using unsupervised learning techniques; training supervised learning models on these labeled datasets; and, finally, using these models to label the remaining RWD data as potentially risky or not. At each step, we provide specific implementation details which can help inform future strategies for identifying potentially risky driving behaviors within RWD data. Thus, our approach first seeks to group observations by driving behavior (i.e., left turns, right turns, accelerating, and merging) and then seeks to identify outlying observations within each group. Further, while researchers and agencies may opt to utilize different specific tools and strategies within each phase of the classification process, the four overarching phases presented herein provide a novel approach for implementing risky driving classification. We note as well that future research should seek to confirm if the process we employ successfully identifies observations related to risky driving outcomes such as car accidents and traffic violations, and we provide recommendations for future steps in the discussion.

2. Data Description and Study Site

Data from the Safety Pilot Model Deployment (SPMD) study were obtained through the Research Data Exchange, via the U.S. Federal Highway Administration (and is now available through Data.gov) [29]. Data were collected during the months of October 2012 and April 2013 in Ann Arbor, MI, from nearly 3,000 vehicles. For this study, data from the first week of April 2013 were utilized and were subsetted to only include data within Washtenaw County (which is, conveniently, in the shape of a rectangle).

This study used basic safety messages (BSMs) transmitted by participating vehicles. BSMs were transmitted at a rate of 10 Hz and contain data on vehicle's state of motion (i.e., speed, acceleration, and yaw rate) and location. Specifically, data from the "BsmP1" file corresponding to April 2013 were used. This file is 204 GB with approximately 1.5 billion observations. For this study, a subset of this file was used corresponding to four weekdays and two weekend days in this first week and contained approximately 268 million observations. Data were stored locally on a PostGreSQL database and were accessed and manipulated using the R programming language. For further details about the "BsmP1" file, the metadata files are referenced [30, 31].

3. Methodology

The overall goal of this study was to design and present a protocol for identifying potentially risky driving behaviors within large RWD datasets. The primary logic of our approach is that the data profile of potentially risky driving

behaviors will look quite similar to the data profile of nonrisky variations of the same behavior (i.e., a risky left turn and a not risky left turn will have similar data profiles) and then that potentially risky behaviors are those which are least normal for its given behavior (i.e., a potentially risky left turn would have a data profile which outlies the average data profile of all left turns in the dataset). As such, the process was divided into four primary stages: reformatting the unlabeled BsmP1 data subsets for analysis (one subset for each day); labeling subsets of the reformatted data as potentially risky or not using the Iterative-DBSCAN (I-DBSCAN) method; using the labeled subsets to train classification models (random forest) for each respective day; and, finally, using the classification models to label the entire day's corresponding data. Random forest was chosen after comparing it with logistic regression and artificial neural networks.

To begin, the BsmP1 data from April 1–7, 2013, were stored in seven different PostGreSQL tables, one for each respective day. Due to a compilation error, the table from Wednesday, April 3, was not included for analysis within this study. As such, the six tables of BsmP1 data corresponding to April 1–2 and 4–7 were utilized. We chose to analyze the data from each day separately for three primary reasons: first, as a matter of feasibility due to the large size of the data files; second, to ensure the reproducibility of the process we employed; and third, because we hypothesize that driving patterns on weekdays versus weekends are likely different (due to work commuting), and thus different types of risky driving behavior may emerge. Regarding the second, we note that consistent reproducibility—while not a reflection of accuracy—is an important feature to establish for any methodological approach. Regarding the third, we generated histograms of observations by time of day for both weekdays and weekends to confirm this hypothesis. Each of these tables (~2–5 GB) was too large to effectively analyze in *R*, and as such, for the first three stages of our process, a random subset of the data (~7–10% of full data) for each of the six days was selected. It was important to ensure that these random samples contained "full driving trips." If we simply pulled random observations, then there would be no guarantee that continuous sequences of observations would be extracted—in the stage one description, the importance of this will be clarified. The BsmP1 data includes unique vehicle IDs and, as such, we randomly selected 100 vehicle IDs for each day (representing ~7–10% of all vehicle IDs) and then extracted all observations corresponding to those vehicle IDs.

3.1. Stage One: Reformatting Subsets. Data were reformatted to address two issues: first, to ensure the data were in a format to best identify potentially risky driving; and second, to reduce the size of the data to improve the runtime feasibility of our labeling method in stage two. Regarding the first, the BsmP1 data are a set of observations measured at a rate of 10 Hz. What is readily apparent when considering these data is that the driving behavior of a vehicle cannot be understood by looking at individual *time-point* observations. A single observation does contain information about speed and acceleration and yaw, but it lacks the context of the full event it is contained within. As such, part of our reformatting process was to take continuous sets of 30 BsmP1 data points and merge them into single observations of *monitoring-period* data representing 3-second windows (30 observations of 10 Hz data correspond to 3 seconds). Regarding the second, these *monitoring-period* observations were generated at one-second intervals (1 Hz), meaning that the reformatted datasets contained 10% of the total number of observations as the original subsets. In Figure 1, we provide a visual depiction of how time-point observations (red diamonds) are converted into monitoring-period observations (blue and green rectangles) for a vehicle moving at a constant velocity—as can be seen, each monitoring-period rectangle contains thirty time period diamonds, with a new monitoring period beginning every ten time period diamonds.

The reformatting process for a single subset was as follows. First, the observations were organized by vehicle ID and then by time. We did not want to combine data corresponding to different vehicles, nor different trips from the same vehicle, so we split each vehicle's data by continuous trip. Since we sorted the data by time as well, we identified the start of new trips by jumps in the recorded time between observations. At this point, the data are divided into individual continuous trips. Then, for each of these trips, the *time-point* observations are merged such that at intervals of one second, three second's worth of data (i.e., thirty observations) were merged into a single observation. The *time-point* data measures of speed, acceleration, yaw, and heading were merged to create *monitoring-period* data measures of average, standard deviation, maximum and minimum values of speed, acceleration, and yaw rate, as well as overall change in heading and standard deviation of change in heading. An array of the unique data identifiers for the 30 *time-point* observations merged was generated as well. The reformatted datasets of *monitoring-period* data were used in the next stage.

3.2. Stage Two: Labeling the Reformatted Data, an Unsupervised Learning Approach. After reformatting, the data were ready to be labeled as potentially risky or not. This task was completed using an unsupervised learning approach, through two primary steps: first, by utilizing k-means clustering algorithm and change in heading thresholds to subset the data into elementary driving behaviors (EDBs); and, second, by utilizing the density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm in an iterative fashion to identify potentially risky driving [32]. The underlying concept behind this approach is that there is a set of EDBs that occurs (such as accelerating, making a U-turn, merging onto the highway, etc.) and that these EDBs will likely have similar statistical profiles to one another. Potentially risky behaviors, then, were identified as the data points which were the further outliers from their prescribed cluster, as identified by running DBSCAN on each EDB cluster—this is meant to capture abnormal instances of EDMs.

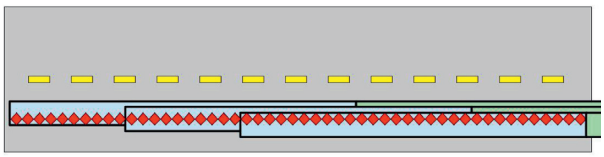


Figure 1: Converting TP data to MP data—using a vehicle moving at a constant velocity as an example. The red diamonds represent TP observations and the blue and green rectangles represent MP observations. Each MP observation contains 30 TP observations and a new MP observation begins every 10 TP observations. Of importance, it is to be noted that there is an overlap in each MP. The fourth, fifth, and sixth monitoring periods are colored green in order to improve visual readability of the figure—the color distinction does not hold further meaning.

The first of these two steps was to identify all EDB clusters within the data. To do this, we first subdivided the data by speed and change in heading. To divide by speed, we ran k-means using only the *average speed* variable to generate three distinct clusters (low, medium, and high speed). The data categorization based on speed has been conducted as a preparatory step in similar previous studies [17, 24]. Then, the data were further subdivided into five different turning classes based on change in heading (left and right turns (change in heading greater than 45 degrees); left and right curves (change in heading between 10 and 45 degrees); and straight (change in heading under 10 degrees)). Subsequently, k-means was run on each of these fifteen subsets, utilizing the sum of squared distances "elbow" method to identify optimal number of clusters (clustering variables were: average, maximum, and standard deviation of speed; average, maximum, minimum, standard deviation, and jerk of acceleration; and, average, maximum, minimum, standard deviation, and jerk of yaw rate). The results of this round of k-means represent the EDB clusters.

For each of the EDB clusters identified, DBSCAN was performed iteratively (I-DBSCAN) [26]. The idea is that, since the data have been clustered into EDBs, the data are dense and that each iteration of DBSCAN will cluster most of the data together. DBSCAN returns n clusters and one set of noise (i.e., unclustered data). One iteration of I-DBSCAN is as follows: first, DBSCAN is run on the dataset—the "elbow" method is utilized to determine the optimal epsilon parameter; second, the "normal" cluster is identified as the cluster consisting of at least 90% of the dataset—if no such "normal" cluster exists, I-DBSCAN is terminated and run again from the beginning; third, all data identified as noise are extracted and labeled as potentially risky; fourth, if any additional clusters have been identified, they are extracted and labeled as potentially risky—if no such additional cluster is identified, then it is checked if this is the third such time no additional cluster has been found and, if so, I-DBSCAN is terminated and the results are returned; finally, if not terminated, another I-DBSCAN iteration is undertaken utilizing the "normal" cluster as the dataset. In a sense, this process is like peeling the layers off of an onion, where the furthest outlying data points are "peeled away" and labeled as potentially risky and the dense set of data in the middle is labeled as not potentially risky. After I-DBSCAN has been run on all the generated EDB clusters, the labeled datasets are merged back together. After running I-DBSCAN on all EDB clusters and merging the results, we have labeled the entire dataset.

In order to complete this entire stage, software is needed to be written to streamline and automate the process. Since the "elbow" method utilized within both k-means and DBSCAN cannot be easily automated, an *R* script was written to semiautomate the labeling process as is described. The script written walked the user through the labeling process, prompting the user to input the values for the "elbow" method when necessary and automating all other aspects of the process.

3.3. Stage Three: Predicting Risky Driving, a Supervised Learning Approach. With the data labeled, the next stage is to train classification models to identify potentially risky driving behaviors. First, it was necessary to identify the optimal classification model to undertake this task. We opted to compare logistic regression, random forest, and artificial neural networks.

3.3.1. Logistic Regression. The logistic regression model is frequently used across the statistical sciences due to both its ease of implementation as well as the ability to extract estimates of causal relationships (in the form of log-odds ratios) [33]. Given a dichotomous outcome Y with possible values of 0 and 1, it is of interest to calculate the probability (as a value from 0 to 1) that an event occurs ($Y = 1$), given a set of known predictors $X = \{x_1, x_2, \ldots, x_n\}$. A typical linear regression model, of which the outcome values range from $-\infty$ to ∞, is not appropriate for modeling dichotomous outcomes [33]. As such, the logistic regression model is defined as follows based upon the logistic distribution:

$$E(Y \mid X) = \frac{e^{\beta_0 + \beta_1 z_1 + \cdots + \beta_n z_n}}{1 + e^{\beta_0 + \beta_1 z_1 + \ldots + \beta_n z_n}}, \tag{1}$$

in which $E\{Y|X\}$ can be understood as the expected value of Y given a set of predictors X [33]. A labeled dataset consisting of dichotomous outcome Y and set of predictors X can be used to fit a logistic regression model, utilizing a maximum likelihood estimator, to calculate model coefficients $\beta = \{\beta_0, \beta_1, \ldots, \beta_n\}$. Once a logistic regression model has been fit, the model can be used to label a dataset consisting m observations of predictors X. For each set of observations, $X_i = \{x_1, x_2, \ldots, x_n\}$, $E(Y|X_i)$ can be calculated, and this value is then assigned to each observation as the prediction of the probability that $Y_i = 1$ [33].

3.3.2. Random Forest. The random forest classification model is a powerful method to implement a form of "ensemble learning," in which many classification trees are generated and whose outputs are aggregated to generate classification predictions [34, 35]. Random forest is built upon the concept of "bagging," in which n classification trees are generated independently of one another, each generated

using a unique bootstrap sample of the training data set [35]. For binary classification, each of the n trees is considered to have a vote, and the final classification of the observation is determined based on majority vote by the n trees. In a standard classification tree, starting from a root node, each node is split based upon all predictors included in the model, but, in random forest, the split decision at each node is made using a random subsample of the available predictors [35]. As noted by Liaw and Wiener, "this somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines, and neural networks, and is robust against overfitting" [35].

As such, given a dichotomous outcome Y with possible values of 0 or 1 and the training set of m vectors of predictors X, n classification trees are generated through the method described above. After being trained, predictions are generated as follows: each of these trees, f_t, given a new set of predictors X', returns a value of either 0 or 1, denoted as $f_t(X') = \{0, 1\}$. The result from each individual tree is considered a vote. The result, either 0 or 1, which gets the most votes, V, is returned as the predicted value Y' for the set of predictors X'. This can be understood mathematically as follows:

$$V = \frac{1}{n}\sum_{i=1}^{n} f_i(X') \longrightarrow Y' = \begin{cases} V < 0.5, & 0 \\ V \geq 0.5, & 1 \end{cases}. \tag{2}$$

3.3.3. Artificial Neural Network. Artificial neural networks arose in response to a digital conundrum: computers are able to solve mathematical computations at a rate that far exceed human capacity, but, simultaneously, cannot solve complex problems that humans are able to do so instantaneously [36]. The overarching concept is that the neural architecture of the human brain is well designed for answering complex questions, and as such, an algorithm replicating this architecture can similarly answer. For this project, we considered a feed forward single hidden layer neural network [37]. In such an architecture, there are three layers of neurons: the input layer, hidden layer, and output layer. The input layer corresponds to the input variables (i.e., one neuron for each variable). Each variable in the input layer is connected by a weighted flow, w, to each of the hidden layer neurons [37]. We used a grid-search approach to determine the optimal number of hidden layer neurons by ranging from 1 to the number of neurons in the input layer. Each of the hidden layer neurons is connected by a weighted flow, β, to the single output layer neuron [37]. As such, given n input variables $X = \{x_1, x_2, \ldots, x_n\}$, m hidden neurons, dichotomous outcome Y, and linear activation function g, the neural network can be defined as follows:

$$G(X) = \sum_{i=1}^{m} \beta_i g(w_i \cdot X + b_i), \tag{3}$$

where $w_i = (w_{i1}, w_{i2}, \ldots, w_{in})^T$ is the vector of flows connecting the n input neurons to the ith hidden neuron, β_i is the flow connecting the ith hidden neuron to the single output neuron, and b_i, is the bias associated with the ith hidden neuron [37]. Given a sample with L total observations, each with predictor sets X_i and dichotomous outcome Y_i, the values of w_i, β_i, and b_i are found by minimizing the distance between the model output and the actual outcome value, as follows [37]:

$$\sum_{i=1}^{L} G(X_i) - Y_i. \tag{4}$$

3.3.4. Evaluating Best Model Fit. In order to evaluate which of these three modeling approaches is best suited for predicting potentially risky driving behaviors, we ran 5-fold cross validation on the labeled subsets. In this process, the dataset is split into 5 groups. For each combination of four groups, the selected four groups are used to train the classification model and then we assess how well the model does at identifying potentially risky driving within the fifth group. The true positive rate and false positive rate of each iteration are calculated in order to create our primary evaluation metric, the area under the receiver operating curve (AUC). We repeated these 5-fold validations 25 times for each of the three classification models and extracted the average AUC scores and corresponding receiver operating curves. As a secondary outcome, runtime was extracted as well. As shall be discussed in the results, the random forest classification model outperformed others.

After it was determined that random forest was the best choice of classification model, a random forest model was fit for each of the six days of data (April 1–2 and 4–7).

3.4. Stage Four: Labeling All the Data. As the random forest models for each of April 1–2 and 4–7 were trained on subsets of BsmP1 data from each of those days, the random forests models were then used to label all of the data in each of these datasets. To do this, data were extracted from each of these datasets by vehicle ID, converted into monitoring-period data format (using the same procedure described in stage one), and then labeled utilizing the respective random forest model. These labeled datasets were then saved in the database by day. At this point, all of the BsmP1 data, reformatted into *monitoring-period* format, for April 1–2 and 4–7, were labeled as potentially risky or not. Since each *monitoring-period* observation included a reference to the 30 *time-point* observations merged to created it, the option is also then available to label the original BsmP1 observations as potentially risky or not (risky if they appear in any *monitoring-period* observations labeled as risky). As an additional analysis, we labeled each daily dataset with each of the other 5 random forest models (i.e., we labeled the April 1 dataset with each of the April 2 and April 4–7 datasets). We then calculated the proportion of the potentially risky observations observed by the daily model (i.e., the April 1st model labeling the April 1st dataset), which are also identified as risky by each of the other day's models. Finally, to better characterize differences between observations labeled as potentially risky and those that are not, we generated

histograms of the distribution of two variables: acceleration jerk (derivative of acceleration) and yaw jerk (derivative of yaw). These values were calculated by comparing the first and last time point of each monitoring period. These variables were chosen because we hypothesize that risky driving behaviors will often be characterized by sudden changes in movement, which may be captured by changes in yaw and acceleration. Given large size of the datasets, we present the histograms with data corresponding to April 1.

4. Results

BsmP1 data were subsetted by calendar day, with a total of six subsets corresponding to April 1–2 and 4–7, 2013 (see Table 1 for number of data points in each table and corresponding number of vehicles). For analysis, 100 vehicle IDs were randomly selected from each day and all data corresponding to each vehicle ID and respective day were extracted (see Table 1 for size of 100 vehicle random sample). Due to technical database issue, the data corresponding to April 3 was not used. We had hypothesized, as well, that weekday and weekend driving patterns would be distinct, with weekday driving patterns being defined by peak driving activity during the morning and evening. In Figure 2, we show histograms of weekday and weekend driving observations by time of day, confirming this hypothesis.

4.1. Stage One: Reformatting the Data. Each of the six subsets was converted from *time-point* observations into *monitoring-period* format. This resulted in the size of the datasets being reduced by an order of magnitude (see Table 2 for number of observations in each table before and after conversion, as well as the number of distinct continuous driving trips identified within each sample).

4.2. Stage Two: Labeling Subsets with I-DBSCAN. The clustering protocol described was applied separately to each of the size reformatted datasets to label all points as either potentially risky or not. The proportion of each dataset labeled as potentially risky ranged from 8.25% to 10.0%, indicating that the clustering protocol behaved in a consistent fashion (see Table 3 for the crude number of data points and the proportion of data points labeled as potentially risky in each dataset).

4.3. Stage Three: Fitting Random Forest Models. With the labeled data in hand, we then compared the performance of three different classification models at correctly identifying potentially risky driving points using 5-fold cross validation. Overall, we found that random forest outperformed both logistic regression and artificial neural network (see Figure 3 for AUROC of each model and Table 4 for mean AUC score and runtime of each classification model).

After identifying random forest as the best classification model, we fit distinct random forest models to each of the six labeled datasets. These random forest classification models correspond to each of the six days.

4.4. Stage Four: Labeling All the Data. The six random forest models fitted in the prior stage were then used to label all of the data in the PostGreSQL database corresponding to the same day. Data were extracted by day and by vehicle, reformatted into *monitoring-period* structure, labeled using the corresponding random forest model, and then inserted into a new PostGreSQL table corresponding to the date of the observation. Table 5 shows the size of the original database tables, the size of the new reformatted, labeled tables, and the proportion of the entries labeled as potentially risky. In Figure 4, we present two heat maps corresponding to data from 250 randomly selected vehicles: one of all observations for these vehicles (left) and the other of the observations labeled as potentially risky.

Next, we sought to determine the performance of cross-applying each random forest model on each of the other datasets. In Table 6, we present the proportion of potentially risky driving behaviors that the same-day model originally found that the cross-day model also found. For example, the April 6 random forest model labeled 223,075 of the April 6 observations as potentially risky—the April 5 random forest model also labeled 72.6% of those 223,075 observations as potentially risky. Overall, the cross-day model always labeled at least 46.6% (ranging up to 80.2%) of the observations that the same-day model had labeled as potentially risky. This provides an indication that different potentially risky driving events occur across different days, and thus separate-day model training seems to be capturing those differences. There appears to be substantial variations by model and day, and thus future research efforts should seek to better understand these variations and improve upon them.

Finally, we sought to characterize differences between potentially risky and not potentially risky driving observations. We hypothesized that some risky driving events would be characterized by more sudden changes in motion and, thus, that the change in acceleration (acceleration jerk) and in yaw rate (yaw jerk) would, on average, be greater than that on nonrisky events. To assess this, in Figure 5, we present histograms of the distribution of the logarithm of acceleration and yaw jerk for both potentially risky and not potentially risky observations from April 1. Plots indicate that risky driving observations tended to be characterized by greater yaw and acceleration jerks. Given the hypothesis that risky driving behaviors are often characterized by sudden changes in movement, this provides initial validation that our approach appropriately identified such observations.

5. Discussion

Here we have presented a multistage process for taking a large, unlabeled RWD dataset and identifying observations representing potentially risky driving behaviors. Modern technological advancements have made bountiful data accessible to transportation researchers, but approaches and solutions to work with these data are requisite if we are to make meaningful improvements to transportation safety. We have shown how unsupervised learning methods—k-means, DBSCAN, and principal component analysis—and supervised learning

Table 1: Subsetting the BsmP1 data.

Date	Database size[1]	Number of vehicles	100-vehicle sample size[1]
Mon, April 1, 2013	44.5	1,395	3.61
Tue, April 2, 2013	51.4	1,418	3.03
Thu, April 4, 2013	50.0	1,430	3.27
Fri, April 5, 2013	50.0	1,405	2.97
Sat, April 6, 2013	39.7	1,133	3.37
Sun, April 7, 2013	32.6	1,072	3.14

[1]Number of observations, in millions.

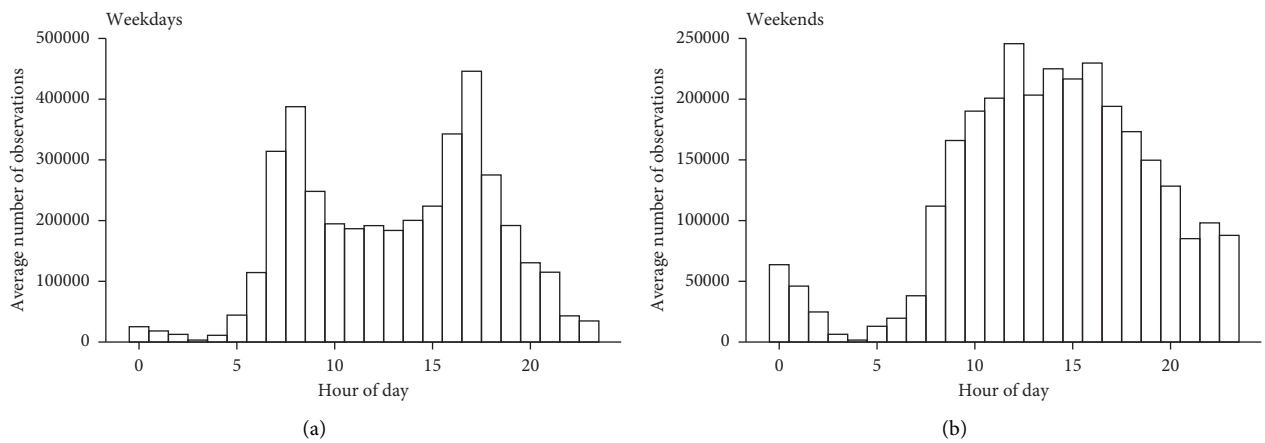


Figure 2: Histograms of observations by time of day for both weekdays (a) and weekends (b).

Table 2: Reformatting the data.

Date	Dataset size prior to conversion	Dataset size after conversion	Distinct vehicle trips
April 1, 2013	3.61 million	291,155	1,383
April 2, 2013	3.03 million	257,752	1,350
April 4, 2013	3.27 million	277,634	3,085
April 5, 2013	2.97 million	250,467	1,225
April 6, 2013	3.37 million	203,073	1,773
April 7, 2013	3.14 million	212,488	811

Table 3: Labeling risky driving data points.

Date	Potentially risky data points	Proportion of dataset (%)
April 1, 2013	24,021	8.25
April 2, 2013	23,063	8.95
April 4, 2013	26,296	9.5
April 5, 2013	25,227	10.0
April 6, 2013	19,672	9.69
April 7, 2013	19,666	9.26

methods—logistic regression, random forests, and artificial neural network—may be applied in a systematic fashion to identify potentially risky driving behaviors within RWD data. While not all RWD datasets will be structured identically, the four stages and details of their implementation provide transportation researchers and professionals the framework necessary to replicate this process and identify potentially risky driving within their own datasets.

While the process defined provides a procedure to identify potentially risky driving behaviors, there are immediate barriers to implementation that must be addressed if such a method is to be made more universally available. In order to undertake the stages as defined, our research team developed software tools in R. DBSCAN, principal component analysis, and k-means all require human interface to identify function parameters (via the "elbow" method), and given that these algorithms needed to be run many times, software which streamlined this process for our team aided in completing this project. As such, there is a need for software solutions which streamline the risky driving identification process. The steps outlined in this paper provide a

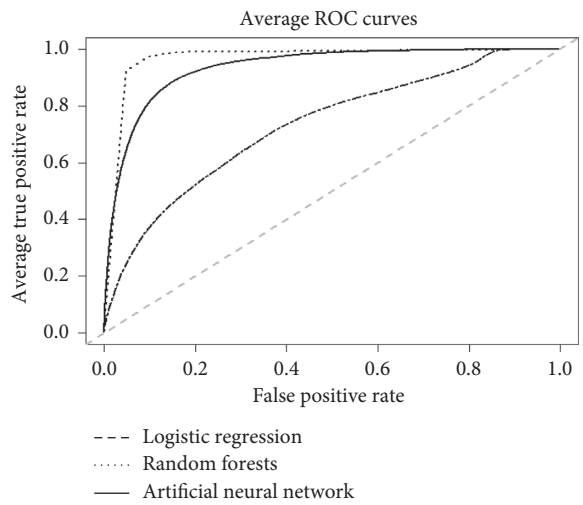


Figure 3: Mean ROC curves for 5-fold cross validation using logistic regression, random forest, and artificial neural network.

Table 4: Mean AUC score and runtime.

Model	Mean area under ROC curve (AUC)	Runtime for single 5-fold iteration (s)
Logistic regression	0.731	7.3
Random forest	0.982	87.6
Artificial neural network	0.927	483.0

Table 5: Risky driving data propositions.

Date	Original database size	Labeled, reformatted database size	Proportion labeled potentially risky (%)
April 1, 2013	44.5 million	3.92 million	7.10
April 2, 2013	51.4 million	4.32 million	7.54
April 4, 2013	50.0 million	4.60 million	7.93
April 5, 2013	50.0 million	4.47 million	8.90
April 6, 2013	39.7 million	2.92 million	7.62
April 7, 2013	32.6 million	2.43 million	6.89

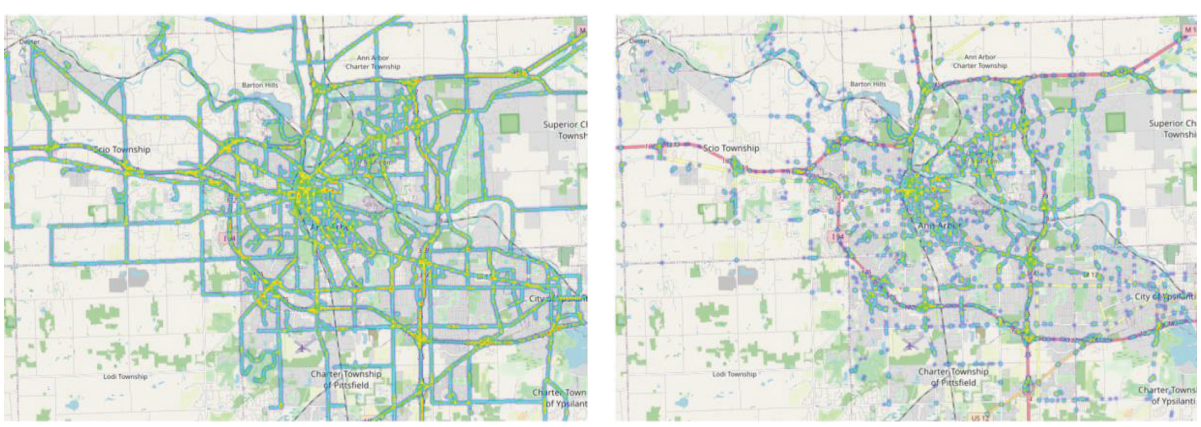


Figure 4: (a) Heatmap of all observations for 250 randomly selected vehicles. (b) Heatmap of all of these observations that were labeled as potentially risky.

novel approach for the implementation of such software solutions.

The applications of this method are immediate. By identifying potentially risky driving behaviors in RWD data, we can identify when and where potentially risky driving behaviors are most concentrated. This will provide transportation agencies real-time, actionable information to improve traffic safety within their given jurisdictions. It also

Table 6: Cross-classifying potentially risky driving behaviors *.

		Dataset labeled					
		April 1 (%)	April 2 (%)	April 4 (%)	April 5 (%)	April 6 (%)	April 7 (%)
Random forest models	April 1, 2013		49.1	65.7	47.1	46.6	52.8
	April 2, 2013	52.0		51.8	69.2	67.6	72.9
	April 4, 2013	59.3	49.2		47.7	50.0	57.0
	April 5, 2013	56.3	73.6	56.4		72.6	80.2
	April 6, 2013	50.6	69.0	54.4	69.4		73.4
	April 7, 2013	50.4	65.7	53.7	68.8	66.8	

*Percentages represent the proportion of the originally labeled observations (by the same-day model) that the cross-day model also identified. We note that all cross-classifications labeled a similar proportion of each dataset as potentially risky (~5–10%).

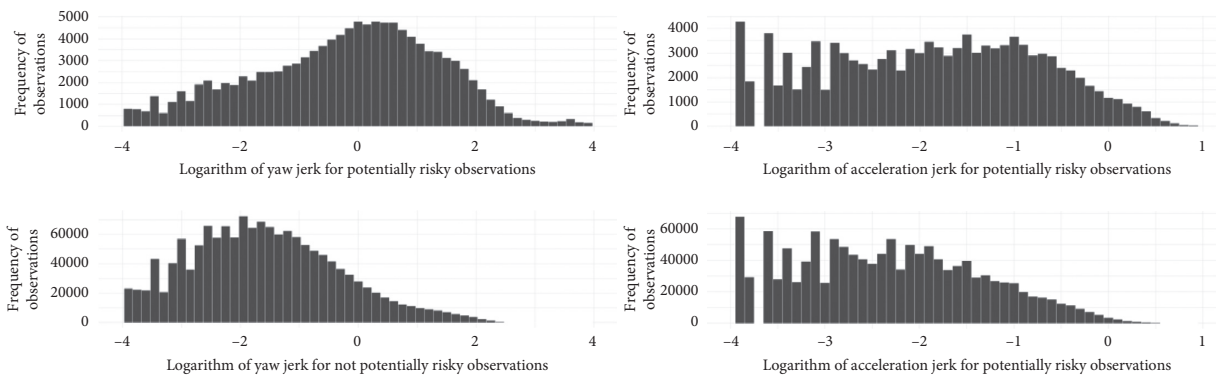


Figure 5: Histograms of the logarithm of yaw jerk (left) and acceleration jerk (right) for both potentially risky labeled observations (top) and not potentially risky labeled observations (bottom) for all observations recorded on April 1.

provides a way to measure the effectiveness of safety countermeasures (i.e., how much risky driving has been reduced after implementation of a desired countermeasure).

A primary limitation of this work is in regard to whether we have truly identified risky driving behaviors or not. The general idea is that, through k-means, we have identified clusters of each elementary driving behavior (EDB) and that potential risky driving points, identified using DBSCAN, are those observations which outlie their given cluster. We have assumed that risky driving behaviors will appear similar to their nonrisky counterparts (i.e., the macro-profile of a nonrisky left turn and a risky left will be very similar), but that when comparing observations of the same EDB, those risky driving behaviors will be identifiable by outlying statistics (i.e., a risky left turn may be identified by a greater acceleration than the average left turn). Future research steps should be taken to assess the external validity of the findings of this method. While we displayed that on average potentially risky driving observations labeled by our approach were characterized by higher yaw and acceleration jerk, future research should also seek to characterize individual EDB to better understand how the statistical profiles of potentially risky data points differ from those not labeled as such. Another limitation of the study was that the models developed were dependent on specific days. Separate-day models were trained, and it was shown that a model trained using a specific day can capture a minimum of 46.6% (up to 80.2% depending on the day) of potentially risky driving events on a different day. This raises a practical consideration in real-world use cases. Future work could focus on developing models for specific days (e.g., Mondays) across different weeks and investigate if, for example, a Monday model could consistently identify different potentially risky events if tested on a different Monday. A hypothesis to explore is that risky driving events are different (to some degree) across different days (i.e., Monday vs Friday) of week but very similar across same days of different weeks (Monday week 1 vs Monday week 2).

6. Conclusion

Overall, this study provides multiple contributions to the advancement of risky driving classification. The overarching steps outlined provide a novel approach by which RWD data can be formatted for and how unsupervised and supervised machine learning methods can be applied to the identification of potentially risky driving behaviors. Further, we have shown specifically how k-means, DBSCAN, and random forests may be applied in this endeavor. We evaluated the predictivity of random forests (in addition to logistic regression and artificial neural network), finding it to be highly sensitive and specific in predicting potentially risky driving behaviors. In sum, we have provided a meaningful process for the implementation of a risky driving classification program, a necessary tool in the efforts to improve traffic safety globally.

Data Availability

The data used to support the findings of this study are publicly available at https://catalog.data.gov/dataset/safety-pilot-model-deployment-data.

Disclosure

The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

Conflicts of Interest

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

Acknowledgments

This research was funded by the Safety through Disruption (Safe-D) National University Transportation Center (UTC), a grant from the U.S. Department of Transportation's University Transportation Centers Program (Federal Grant Number: 69A3551747115).

References

[1] World Health Organization, *Global Status Report on Road Safety 2018*, World Health Organization, Geneva, Switzerland, 2018.

[2] National Highway Traffic Safety Administration, *Aggressive Drivers View Traffic Different Capital Beltway Focus Groups Find*, National Highway Traffic Safety Administration, Washington, DC, USA, 1998.

[3] C. Ma, W. Hao, W. Xiang, and W. Yan, "The impact of aggressive driving behavior on driver-injury severity at highway-rail grade crossings accidents," *Journal of Advanced Transportation*, vol. 2018, pp. 1–10, 2018.

[4] T. E. Boyce and E. S. Geller, "An instrumented vehicle assessment of problem behavior and driving style:," *Accident Analysis & Prevention*, vol. 34, no. 1, pp. 51–64, 2002.

[5] B. G. Simons-Morton, Z. Zhang, J. C. Jackson, and P. S. Albert, "Do elevated gravitational-force events while driving predict crashes and near crashes?," *American Journal of Epidemiology*, vol. 175, no. 10, pp. 1075–1079, 2012.

[6] S. Klauer, T. Dingus, V. Neale, J. Sudweeks, and D. Ramsey, *Comparing Real-World Behaviors of Drivers with High versus Low Rates of Crashes and Near Crashes*, National Highway Traffic Safety Administration, Washington, DC, USA, 2009.

[7] L. Evans, *Traffic Safety*, Science Serving Society, Bloomfield Hills, MI, USA, 2004.

[8] R. Paleti, N. Eluru, and C. R. Bhat, "Examining the influence of aggressive driving behavior on driver injury severity in traffic crashes," *Accident Analysis & Prevention*, vol. 42, no. 6, pp. 1839–1854, 2010.

[9] AAA Foundation for Traffic Safety, *Aggressive Driving: Research Update*, AAA Foundation for Traffic Safety, Washington, DC, USA, 2009.

[10] M. H. Parry, *Aggression on the Road: A Pilot Study of Behaviour in the Driving Situation*, Tavistock Publications, London, UK, 1968.

[11] L. Mizell, M. Joint, and D. Connel, *Aggressive Driving: Three Studies*, AAA Foundation for Traffic Safety, Washington, DC, USA, 1997.

[12] D. Shinar, "Aggressive driving: the contribution of the drivers and the situation," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 1, no. 2, pp. 137–160, 1998.

[13] K. H. Beck, M. Q. Wang, and M. M. Mitchell, "Concerns, dispositions and behaviors of aggressive drivers: what do self-identified aggressive drivers believe about traffic safety?," *Journal of Safety Research*, vol. 37, no. 2, pp. 159–165, 2006.

[14] S. K. Balogun, N. A. Shenge, and S. E. Oladipo, "Psychosocial factors influencing aggressive driving among commercial and private automobile drivers in Lagos metropolis," *The Social Science Journal*, vol. 49, no. 1, pp. 83–89, 2012.

[15] L. Tasca, "A review of the literature on aggressive driving research," in *Proceedings of the First Global Web Conference on Aggressive Driving*, Ontario, Canada, 2000.

[16] F. Feng, S. Bao, J. R. Sayer, C. Flannagan, M. Manser, and R. Wunderlich, "Can vehicle longitudinal jerk be used to identify aggressive drivers? An examination using naturalistic driving data," *Accident Analysis & Prevention*, vol. 104, pp. 125–136, 2017.

[17] X. Wang, A. J. Khattak, J. Liu, G. Masghati-Amoli, and S. Son, "What is the level of volatility in instantaneous driving decisions?," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 413–427, 2015.

[18] J.-H. Hong, B. Margines, and A. K. Dey, "A smartphone-based sensing platform to model aggressive driving behaviors," in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, pp. 4047–4056, Denver, CO, USA, 2014.

[19] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong, and M. Li, "Fine-grained abnormal driving behaviors detection and identification with smartphones," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2198–2212, 2017.

[20] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, 2020.

[21] D. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 1609–1615, Washington, DC, USA, 2011.

[22] A. Jahangiri, V. J. Berardi, and S. Ghanipoor Machiani, "Application of real field connected vehicle data for aggressive driving identification on horizontal curves," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2316–2324, 2018.

[23] M. H. Alkinani, W. Z. Khan, and Q. Arshad, "Detecting human driver inattentive and aggressive driving behavior using deep learning: recent advances, requirements and open challenges," *IEEE Access*, vol. 8, pp. 105008–105030, 2020.

[24] A. Jahangiri, S. G. Machiani, and V. Balali, "Big data exploration to examine aggressive driving behavior in the era of smart cities," in *Data Analytics For Smart Cities*, pp. 163–182, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2019.

[25] J. Lee and K. Jang, "A framework for evaluating aggressive driving behaviors based on in-vehicle driving records," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 65, 2017.

[26] C. Marks, A. Jahangiri, and S. Ghanipoor Machiani, "Iterative DBSCAN (I-DBSCAN) to identify aggressive driving

behaviors within unlabeled real-world driving data," in *Proceedings of the 22nd Intelligent Transportation Systems Conference*, Auckland, NZ, USA, 2019.

[27] R. Akikawa et al., "Smartphone-based risky traffic situation detection and classification," in *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 1–6, Austin, TX, USA, 2020.

[28] M. Jeihani, A. H. Pour, and A. Ardeshiri, *Machine Learning Model for Driving Distraction Detection*, Morgan State University, Baltimore, MD, USA, 2020.

[29] US Department of Transportation, *Safety Pilot Model Deployment Data*U.S. Department of Transportation, Washington, DC, USA, 2018.

[30] US Department of Transportation, *Safety Pilot Model Deployment–Sample Data, from Ann Arbor, Michigan, Version 1*, U.S. Department of Transportation, Washington, DC, USA, 2014.

[31] US Department of Transportation, *Safety Pilot Model Deployment Sample—Data Environment Data Handbook, Version 1.3*, U.S. Department of Transportation, Washington, DC, USA, 2015.

[32] M. Ester and H.-P. Kriegel, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, AAAI Press, New Orleans, LA, USA, 1996.

[33] D. Hosmer, S. Lemeshow, and R. Sturdivant, *Applied Logistic Regression*, Wiley, Hoboken, NJ, USA, 3rd edition, 2013.

[34] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2-3, 2002.

[36] A. K. Jain, J. Jianchang Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[37] F. Lolli, R. Gamberini, A. Regattieri, E. Balugani, T. Gatos, and S. Gucci, "Single-hidden layer neural networks for forecasting intermittent demand," *International Journal of Production Economics*, vol. 183, pp. 116–128, 2017.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 6634944, 14 pages
https://doi.org/10.1155/2021/6634944

Research Article

Model-Based Predictive Detector of a Fire inside the Road Tunnel for Intelligent Vehicles

Marián Hruboš,[1] Dušan Nemec,[1] Emília Bubeníková,[1] Peter Holečko,[1] Juraj Spalek,[1] Michal Mihálik,[1] Marek Bujňák,[1] Ján Andel,[1] and Tomáš Tichý[2]

[1]*Department of Control and Information Systems, Faculty of Electrical Engineering and Information Technology, University of Žilina, Žilina, Slovakia*
[2]*Department of Transport Telematics, Faculty of Transportation Scieneces, Czech Technical University, Prague, Czech Republic*

Correspondence should be addressed to Marián Hruboš; marian.hrubos@feit.uniza.sk

Received 8 October 2020; Revised 26 January 2021; Accepted 2 February 2021; Published 12 February 2021

Academic Editor: Petr Dolezel

The paper proposes a method for detection of a fire inside the road tunnel without direct view on the fire, using on-board vehicle technologies. The system is based on comparing the measured development of temperature and smoke with model scenarios precomputed for a given road tunnel. The fire scenarios are computed by HW/SW tool TuSim regarding the parameters of the real road tunnel and then the results are presented to the vehicles via car-to-infrastructure communication link. The proper detection of the fire allows early evacuation of the vehicle passengers, which will significantly increase chance of their survival. The computed scenarios also provide supporting information for the rescue teams.

1. Introduction

Safety of transported persons and material is an integral part of toady's transport. From the viewpoint of risk and subsequent damage, the worst place is the road tunnel. In case of accident, several dangerous situations occur in the tunnel. One of the most dangerous accidents at all is the fire inside the tunnel. Even when the fire is detected by sensors and cameras installed in the tunnel tube, statistics says that not all passengers in the threatened zone will evacuate from the vehicles in time. We believe that this drawback can be suppressed when not only the operators of the tunnel but also the vehicles themselves are sensing and detecting the indicators of the fire-rising temperature and opacity.

Subsequent recovery after the accident is also challenging and can expose other persons to danger. Therefore, we propose a simulation model that is able to estimate situation in the tunnel during an accident event. Furthermore, such information can be utilized by rescue teams.

The paper deals with an effect of the technological equipment on safety of humans and property in the tunnel. To reach a tolerable level of safety, the heterogeneous complementary systems must be installed in the tunnel. The numerical expression of the risk is problematic. Therefore, simulation is one of few possibilities of how to safely compare various variants and test the system limits, so-called worst cases, as an alternative to the prescriptive risk analyses. This paper is focused on the simulation of critical scenarios itself, on the creation of trustworthy models and their subsequent verification and validation.

Nowadays, the tunnel simulators may be classified to the following groups:

(1) Simulators used to train tunnel operators, dealing mainly with virtual reality or tunnel visualization [1].

(2) Drive simulators used to train drivers driving through the tunnel; they make it possible to monitor and analyse drivers' behaviours, technical parameters of their drives, minds, subjective feelings, etc. [2].

(3) Specialized simulation tools such as IDA RTV software [3].

(4) Simulators based on the PLC (programmable logic controller) that are mostly used to verify control of tunnel technologies under various modes before putting the control system into operation [4, 5]. These simulators may utilize additional specialized tools for simulation of the tunnel technology and physics; an architecture of such a simulator is depicted in Figure 1.

Fire, in general, can be detected by the following way:

(1) SOS button

(2) Video detection

(3) Smoke sensors

(4) FibroLaser line detector

In principle, video detection can detect a fire in several ways, by detecting a car/vehicle stop, fire, or smoke in the monitored zone. In time, this method is the fastest method; on the other hand, false alarms can occur from the smoke in tunnel through the passage of truck or light reflection. In this case, in Slovakia the tunnel is usually closed only after approval by the operator. The detection of stopped vehicle is almost immediately. There are two types of smoke sensors: opacity sensors of specialized smoke sensors. Opacity sensors must be installed closed to portal, branching at maximum distance of 1000 meters along tunnel. According the manufacture's materials SIGRIST, the opacity sensors have a measuring range of 0–100 km; accuracy in the range 0–15 km must be ± 2 km. These sensors can be also used for fire detection; they are usually installed on the wall of the tunnel. Smoke sensors must be installed at the maximum distance of 150 meters along tunnel. The measure range of these smoke sensors is 0–15 km and accuracy is 0, 2 km; they are usually installed on the tunnel ceiling.

Typical values in measuring opacity:

(1) Normal traffic <5 km

(2) Heavy traffic ~5 km

(3) Traffic jam ~7 km

(4) Tunnel close 12 km

(5) Fire >15 km

FibroLaser detects an increased heat in fire in the tunnel. It is built up of optical cables and control unit that emits a laser beam into the cable and analyses its reflection. There is a Raman effect, when the reflected laser beam is divided into Stokes and AntiStokes signal and temperature change in optic cable is evaluated based on the difference in the intensity of these signals. It is possible to implement FibroLaser into simulation programs according the data of the manufacturer Siemens. There are three rules for how to detect the increase in temperature:

(1) Overrun defined maximum value

(2) Overrun the maximum difference from the average temperature zone

(3) Overrun the maximum increase of temperature in define time

2. Tunnel Simulator TuSim

The tunnel simulator (TuSim), developed by the authors of the article, is based on the programmable logic controller (PLC). The TuSim is a complex HW/SW solution based on the industrial personal computer (PC), Bernecker and Rainer (BR) Automation PC acting as a PLC. The TuSim is shown in Figure 2 and consists of (top-down view) the Masterview liquid-crystal-display (LCD) switch of the visualization server, BR Automation PC (in the left bottom), and the backup UPS unit (in the right bottom). The uninterruptible power supply (UPS) unit ensures simulation of the continuous operation.

In the BR Automation PC, the PLC of Siemens S7-400 type is simulated including the technological components as well. The TuSim simulator makes it possible to simulate three types of the tunnel, 1000 m long each:

(1) Urban tunnel (MST)

(2) Highway one-tube tunnel (D1T)

(3) Highway two-tube tunnel (D2T)

Visualization of technological equipment is ensured through the human-machine interface/supervisory control and data acquisition (HMI/SCADA) displays. The whole system has an open software concept for future extensions from the level of software in the PLC up to the design of graphical screens.

The simulator is not connected to the data flow of a real tunnel. The HMI server takes care of data collection, archiving, and distribution from PLC clients. Selection of data to be archived in the database at the server is configured by the Database Logger [6], and then data may be shown in all clients. In addition to data display, the clients also make it possible to perform control interventions in displays individually for each technological sub-system. To display and change the screens, the tools CimViewer and CimEdit [7] from the HMI/SCADA CIMPLICITY software are being used.

For the reason of model verification, the PLC may be interconnected to other tools in the following ways:

(1) Extensible markup language (XML) interface provided by the ELTODO company

(2) Libnodave dynamic-link library (DLL library) [8]

(3) MATLAB/SIMULINK [9]

(4) IDA road tunnel ventilation (RTV) [3]

3. Simulation Models

In the first version, the TuSim was a drive simulator helping service operators to become familiar with the control system of the tunnel and to simulate emergency events manually via so-called reflexes. Since no mathematical-physical models, needed to simulate functionality of the many tunnel systems, were built-in, they had to be integrated additionally. Figure 3 shows models important for basic functionality and interactions between them. The models in blue fields are implemented directly on the PLC level, others on the HMI/SCADA level.

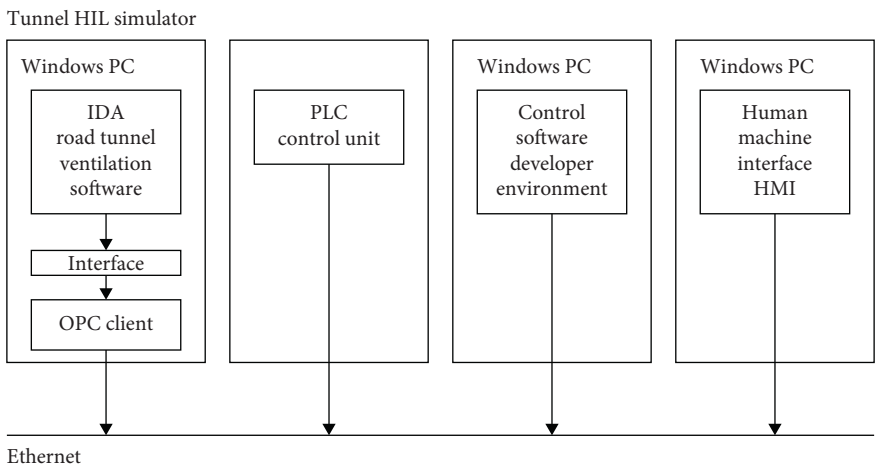


Figure 1: Architecture of the PLC-based Hardware-in-the-Loop (HIL) tunnel simulator [5].

Figure 2: TuSim.

3.1. Traffic Flow Model. As it is apparent from Figure 3, the model of air flow processes the piston effect resulting from the movement of vehicles inside the tunnel as one of its inputs. Under the one-way traffic, vehicles put the air to motion. The more significant the contribution of that, the higher the volume of vehicle intensity. Products of combustion in the tunnel are also an important input to control air technology. It is also important to know composition of traffic flow, since trucks and buses have much higher impact on emissions in the tunnel environment. More information about the traffic model of the TuSim is in [10–12].

3.2. Tunnel's Tube Model. The tunnel tube may be modelled by multiple ways. One of them is based on analysis of the relationship between its inputs and outputs. This approach has been used in [13]. To describe a linear part of the model, we used the state model. The non-linear part of the model includes saturation and transport delay. The model of the tunnel tube is expected to enable changing of carbon monoxide CO and nitrous oxides NO_x sensors positions. That will make it possible to monitor variances in ventilation control. Concentration of emissions in time and space may be calculated using the equation for the longitudinal ventilation [14]:

$$\frac{\partial C}{\partial t} + \frac{\partial (vC)}{\partial x} = e_C, \tag{1}$$

where t is time of simulation (s) and x is distance (m).

If the measured and calculated velocity of the air flow in the tunnel is available, we can immediately use solution of the equation from [14], for example, for the values of CO in a steady state:

$$C(x) = \frac{e_C}{v} + C(0),$$
$$e_c = \frac{NE}{3.6 v_v A'}, \tag{2}$$

where

(1) $C(0)$ is the concentration of CO at the input portal of the tunnel (μg/m^3)

(2) v is the velocity of the air flow in the tunnel (m/s)

(3) v_v is the average velocity of the vehicle movement (km/h)

(4) N is the traffic volume (veh/h)

(5) E is the COemissions of the vehicle (g/h.veh)

(6) A is the tunnel cross section (m^2), and A' is a derivation of A

In our model, the values of emissions for all velocities and gradients are stored in the table. Outputs of the model of

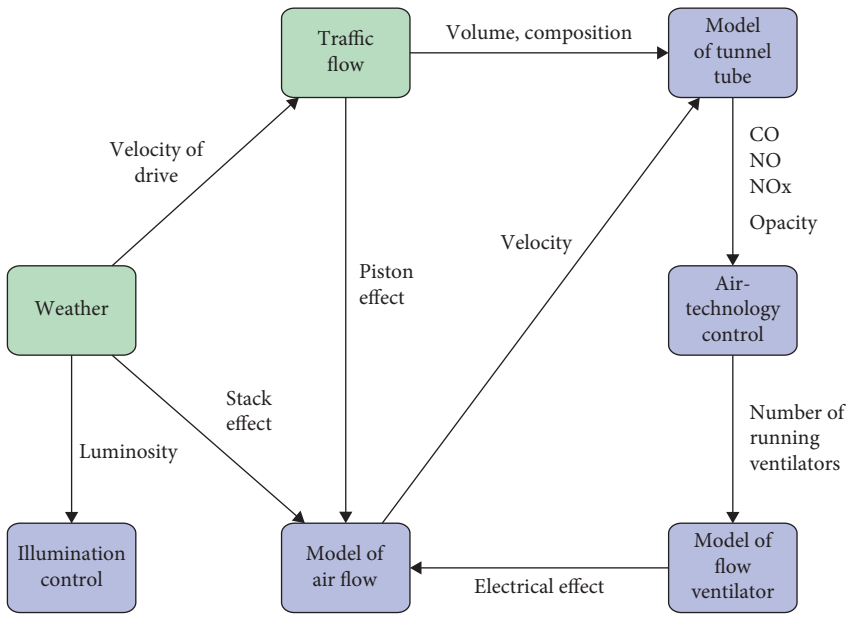


Figure 3: Interactions among modules of the extended simulation model TuSim.

the longitudinal ventilation for one-way traffic will create a linear function, with growing concentration of CO emissions towards the output portal of the tunnel. In [15], there are a discrete microscopic traffic model and emissions modelled for each section occupied by a vehicle according to the tables of emissions [16]. The advantage is that emissions from vehicles do not depend on shape of the tunnel; thus, the model may be used universally. Figure 4 shows comparison of our traffic model extended for the model of the tunnel tube with the model of steady state conditions.

After velocity of the air flow becomes stabilized, the values of CO emissions in the microscopic model are close to the values of emission models of the macroscopic traffic model along the whole length of the tunnel. For the needs of the traffic model, we have adopted the PIARC tables [16] for discrete velocities used in the traffic model – 0 (km/h), 30 (km/h), 50 (km/h), maximum velocity in the tunnel 80 (km/h), maximum gradient 4%.

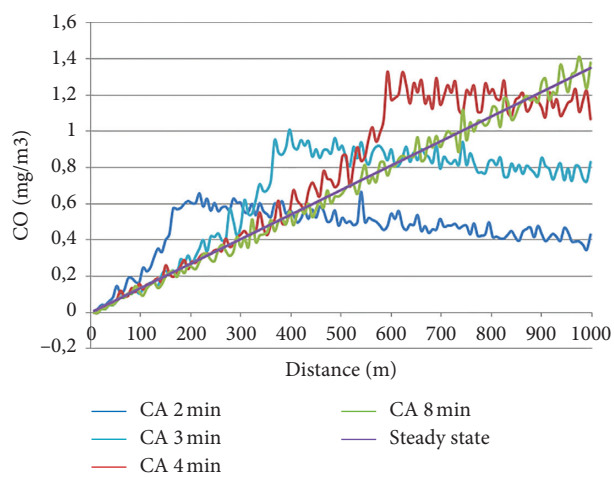


Figure 4: Comparison of models of CO emissions.

3.3. Air Flow Model. Simulation of the air flow in the tunnel is a complex problem demanding numeric solution of Navier–Stokes non-linear differential equations. Their solution is too time demanding to be used in the real-time in the PLC. As an alternative, one might use Bernoulli equation for one-dimensional liquid flow while we consider that the air is as an incompressible fluid:

$$\frac{1}{2}\rho v^2 + \rho g h + p = \text{const.for } p \sim \text{const}, \tag{3}$$

where

(1) ρ is the air density (kg m^{-3})

(2) v is the air velocity (m/s)

(3) p is the pressure (Nm^{-2})

(4) h is the height difference between the ends of the tunnel (m)

(5) g is the gravitational acceleration (ms^{-2})

This formula is valid only in the case of an ideal liquid; in the case of real liquid, the equation must be extended for a member representing friction losses. Velocity of the flow in the tunnel is influenced by many factors: difference in temperatures between the tunnel and outside environment, gradient of the tunnel, piston effect, effect of fans, friction, change of the cross section, weather conditions at the portal, etc. In our case, the majority of Slovak tunnels have non-rugged profiles; i.e., their cross section is the same, and so only one equation must be solved. Otherwise, a system of equations should be solved for each section of the tunnel in the case of the profile change, branching-off inside the tunnel, when the ventilation shaft is used (Branisko tunnel)

or combined ventilation systems (Višňové tunnel being under construction) [17].

In order to compare the methods, we have created a model similar to the Bôrik tunnel in the IDA RTV [6], with circumference $P = 29.22\,\mathrm{m}$ and cross section area $A = 57.26\,\mathrm{m}^2$. To calculate the air flow, we used the substitutionary circular cross section of the tube whose hydraulic diameter D can be calculated [18]:

$$D = \frac{4A}{P} = \frac{4(57.26)}{29.22} = 7.83846\,(\mathrm{m}). \tag{4}$$

Flow fans put air to motion; pressure change depends on a number of fans, efficiency, and fan area. References [19, 20] give multiple versions of the equation for both mobile fans and ceiling flow fans:

$$\Delta P_{\mathrm{FAN}} = \frac{\eta I_{\mathrm{FAN}}}{A}\left(1 - \frac{v}{v_{\mathrm{FAN}}}\right), \tag{5}$$

where

(1) A is the area of tunnel cross section (m^2)
(2) η is the efficiency of the fan (%)
(3) I_{FAN} is the pushing force of the fan (N)
(4) v is the velocity of the air flow (m/s)
(5) v_{FAN} is the velocity of the fan air (m/s)

For one-way traffic, the moving vehicles move air in the tunnel on, creating so-called piston effect. The higher the velocity and traffic volume are, the higher this effect is. Due to emergency situations in the tunnel, we are also interested in the situation with stopped vehicles when their velocity is lower than velocity of air flow in the tunnel. Since air flow decelerates, we used the equation with the absolute value of velocity [21]:

$$\Delta P_{\mathrm{PISTON}} = \frac{\sum_{i=0}^{4} N_i C_i A_i}{2A}\rho\left|v_v - v\right|(v_v - v), \tag{6}$$

where

(1) A is the area of the tunnel cross section (m^2)
(2) ρ is the air density $(\mathrm{kg/m}^3)$
(3) v is the velocity of air flow (m/s)
(4) v_v is the velocity of vehicles (m/s)
(5) N_i is the number of vehicles
(6) A_i is the front area of the vehicle (m^2)
(7) C_i is the friction coefficient

Under fire conditions, temperature in the tunnel will increase which will cause temperature difference between internal temperature and temperature of surrounding environment. The fire represents barrier to air flow. The local loss of pressure caused by the fire depends on temperature power, shape of the lateral cross section of the tunnel, and other factors. The most accurate way of determining temperature in the tunnel is using the CFD simulation or evaluation of real fire experiments. The average temperature in the whole fire section may simplistically be calculated [17]:

$$T_m = T_0 + (T_{\mathrm{fire}} - T_0)\exp\left(\frac{-\alpha P}{v\rho A c_p}x\right), \tag{7}$$

where

(1) ρ is the air density $(\mathrm{kg/m}^3)$
(2) A is the area of the tunnel cross section (m^2)
(3) T_0 is the temperature in front of the place of fire (K)
(4) T_{fire} is the temperature at the place of fire (K)
(5) α is the coefficient of the heat transfer $(\mathrm{W/m^2K})$
(6) P is the circumference of the tunnel (m)
(7) c_p is the specific heat capacity of the air (kJ/(kgK))
(8) x is the distance from the place of fire (m)

The final differential equation of the air flow velocity was obtained as sum of all mentioned pressure differences and also others described in [22]:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{\sum \Delta P}{\rho L}, \tag{8}$$

where

(1) ΔP is the all elements mentioned above (Pa)
(2) ρ is the air density $(\mathrm{kg/m}^3)$
(3) L is the length of the tunnel (m)

Furthermore, other members of the equation, such as the temperature differences and the influence of the wind, can be considered.

3.4. Fire Model. The course of the fire may be simulated in various tools. For the reason of calculation time, the three-dimensional simulation by the fire dynamics simulator (FDS) was excluded. There may be two-dimensional simulation using CFAST [23] taken into account which is primarily designed for simulation of the fire in buildings. CFAST was extended for simulation of air flow in long corridors. In [24], the tunnel consisted of several interconnected corridors for the zone model; the results were compared to FDS. For the empty tunnel and low air flow, the results were comparable. For obstacles in the tunnel and various air flow velocities, we found out it is not possible to use CFAST reliably for fires in the tunnels. Therefore, we used the one-dimensional model of the fire, similarly as in the document TP02/2011 [19], or in IDA RTV [6], with pre-determined curves of the fire power for each type of the vehicle. Shape of the curve for the fire power may be mathematically simplified either linearly, exponentially, or quadratically [20]. For simulation, we chose model curves of fire development according to real tests (Figure 5).

References [25, 26] give equation for calculation of temperature at the place of fire:

$$T_{\mathrm{fire}} = \frac{Q}{v\rho A c_p} + T_0, \tag{9}$$

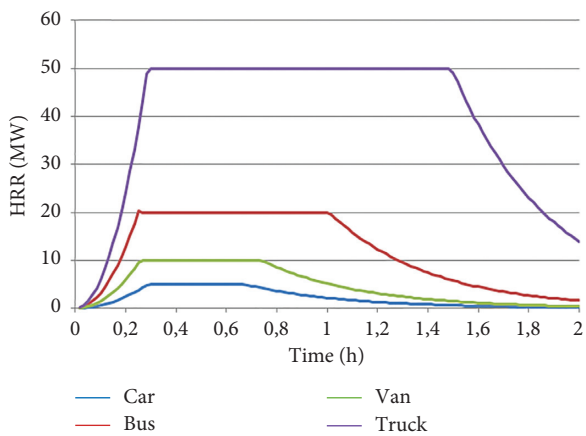


Figure 5: Model curves of the heat power of the fire.

where

(1) Q is the fire power (W)

(2) v is the velocity of air flow (m/s)

(3) ρ is the air density (kg/m^3)

(4) A is the area of cross section of the tunnel (m^2)

(5) T_0 is the outside temperature (K)

(6) c_p is the specific heat capacity of air (kJ/(kgK))

Since we have the one-dimensional model, temperature is considered as an average temperature of the cross section at the place of fire. According to [25], comparison with the three-dimensional computational fluid dynamics (CFD) simulation in situations without backflow of the smoke gives a good coincidence of temperatures (error ca 1%). In the case of backflow of the smoke, an error occurs. Burning efficiency is not 100% so the power in equation must be reduced. We chose the value 90%. Further, we applied only a part of the reduced power of the fire, approximately 70%, in accordance with the references; the residual power is radiated to the wall of the tunnel. For its temperature, we can use the following equation [22]:

$$Q_{\text{wall}} = \varepsilon\delta\left(T_{\text{fire}}^4 - T_{\text{wall}}^4\right)\pi\ DL_{\text{FIRE}} + \alpha\left(T_{\text{fire}}^4 - T_{\text{wall}}^4\right)\pi\ DL_{\text{FIRE}}, \tag{10}$$

where

(1) ε is the emissivity (−)

(2) δ is the Stefan–Boltzmann constant (W/(m^{2}K^4))

(3) α is the coefficient of the heat transfer (W/(m^{2}K))

(4) T_{wall} is the temperature of the wall (K)

(5) T_{fire} is the air temperature (K)

(6) L_{FIRE} is the length of the tunnel with fire (m)

(7) D is the hydraulic diameter of the tunnel (m)

The given equation considers radiation to tunnel walls one-dimensionally; there is difference between the ceiling and the pavement.

Generally, the fire may be detected by multiple ways. For our simulations, there were more important autonomous systems: smoke detection and linear detector FibroLaser [27]. For simulations, we used variable detection time.

3.5. Smoke Propagation Model. Smoke propagation depends on velocity of air flow in the tunnel, size of the fire, cross section of the tunnel, and the tunnel gradient. Smoke whose temperature is higher than temperature of air in the tunnel is propagated below the ceiling and depending on velocity of air flow it propagates one way or both ways. Figure 6 shows possibilities of smoke propagation in the tunnel.

In Figure 6(a), velocity of air flow is low, and smoke stratifies evenly. Fire ventilation in the tunnel for both-way operation should follow this case since there are persons along both sides of place of fire. In Figure 6(b), velocity of air flow is lower than critical velocity, and smoke destratification occurs, backflow propagation of the smoke. In Figure 6(c), velocity of air flow is higher than critical velocity, and smoke destratification does not occur. For one-way traffic, knowledge of critical velocity for fires of various vehicle types is key knowledge for safety of persons in the tunnel. Comparison of various ways of how to calculate critical velocity in dependency on fire power is shown in Figure 7 For the simulation, we chose an analytical calculation [27].

The critical expression of critical speed was chosen for its optimality with respect to all commonly used approaches to the calculation of critical speed. The comparison was made based on the authors' analysis method [29] and calculation using software IDK RTV, TP12/2011.

According to the technical conditions of ventilation of road tunnels TP12/2011, this speed for longitudinal ventilation in the direct of traffic in one-way traffic should be greater than the so-called critical speed at which smoke spreads back. If people are only in one direction from the fireplace, fire ventilation has to be controlled, so that flow speed in higher than or equal to the critical speed.

Calculation of critical speed according to TP12/2011 is as follows:

$$v_{\text{crit}} = C_0 C_3 \sqrt{C_1 C_4}\ \frac{\sqrt{1 + (1 - (C_2/C_1))C_4\left(B^2/gH\right)}}{1 + C_4\left(B^2/gH\right)} B, \tag{11}$$

where input values are

$Q = 5.106$ W for a car fire, the amount of the heat released,

$g = 9{,}81$ m.s^{-2} gravitational acceleration,

cp = 1039 J/(kg.K) heat capacity of the clean air stream,

$H = 6{,}995$ m clearance height of tunnel,

$W = 9{,}5$ m clearance width of tunnel,

$A = 57{,}26$ m^2 area of clearance cross section tunnel,

$s = 1\%$ gradient of tunnel,

$a = 1{,}134$ kg/m3 flow density of clean air,

Ta = 288, 15 K clean air flow temperature.

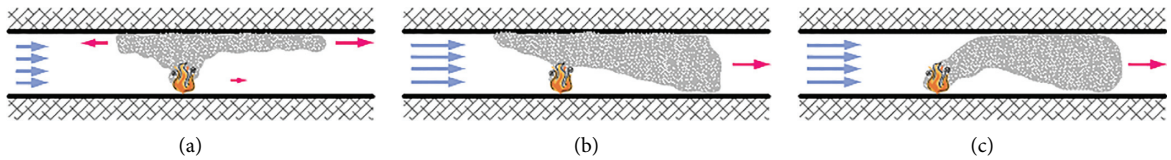


Figure 6: Smoke propagation depending on velocity of air flow [28].

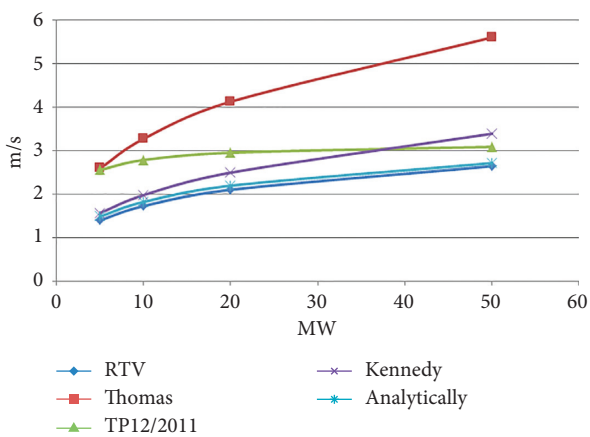


Figure 7: Comparison of various ways of critical velocity calculation.

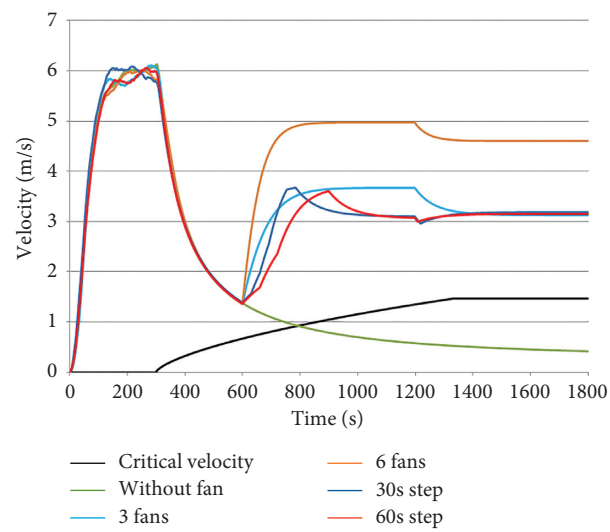


Figure 8: Comparison of algorithms of ventilation control for 5 MW fire.

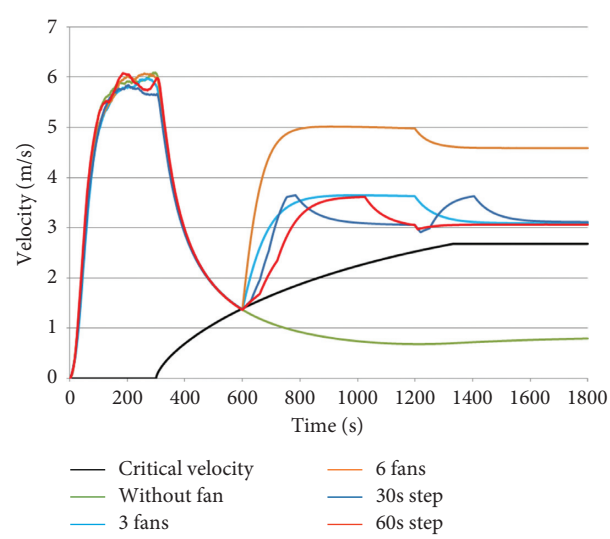


Figure 9: Comparison of algorithms of ventilation control for 50 MW fire.

The constant values of highway double-tube tunnel are $C0 = 0,9$, $C1 = 0,91935$, $C2 = 0,42233$, $C3 = 0,61299$, $C4 = 7,51768$.

For different scenarios of vehicle fire, only coefficient B is changed:

$$B = \sqrt[3]{\frac{QgH}{c_p T_a \rho_a A}}. \tag{12}$$

$$\begin{aligned}
&5\,\text{MW}: B = 2{,}6036,\ \text{vcrit} = 2{,}565\ \text{m/s},\\
&10\,\text{MW}: B = 3{,}2803,\ \text{vcrit} = 2{,}794\ \text{m/s},\\
&20\,\text{MW}: B = 4{,}1329,\ \text{vcrit} = 2{,}961\ \text{m/s},\\
&50\,\text{MW}: B = 5{,}6093,\ \text{vcrit} = 3{,}097\ \text{m/s}.
\end{aligned}$$

3.6. Fire Ventilation Algorithm. We analysed the control algorithm of fire ventilation for traffic volume 1600 (veh/h) and for the values 5 MW and 50 MW of firepower. For comparison, we chose the following ways to control ventilation: switching of all the fans off, switching of three fans on (half of all except for the fire zone), switching of six fans on (all except for the fire zone), and regulation to the required value within the interval from 3 to 3.5 ms^{-1} with the steps 30 s and 60 s. The control system activated the chosen way of ventilation control in the 5th minute from stopping the traffic. The result of comparison for 5 MW is shown in Figure 8, and for 50 MW in Figure 9. In the 20th minute, we suddenly chose velocity of the wind 4 (m/s) in the opposite direction to assess decrease in the flow value. We can see that for deactivated ventilation velocity of the flow decreases under the value of the critical velocity at the time 800 s, i.e., approximately 8 minutes after traffic had been stopped. For 5 MW fire and the fixed number of switched-on fans, the value of velocity of the flow did not decrease below the value 3 (m/s), even for 3 fans. It means that condition of 100 % redundancy of fans was fulfilled. When regulating to the required value of the air velocity, we can see that the step 30 s has a steeper rise to the required value, but a higher overshot as well. The error caused by the wind was in both cases immediately compensated by increase in fire; fire reached its maximum in time of the constant value of critical velocity.

For 50 MW, we can see that reaction of the control system 5 minutes after stopping traffic is already at the critical value of velocity for the given fire. Similarly, as in the previous comparison, only three fans were sufficient to keep the value of air flow 3 (m/s), so again the requirement of 100% redundancy of fans was fulfilled. Higher velocity of air flow guarantees lower temperature of air in front of the fire place, ensures delivery of fresh air, and keeps smoke in direction from evacuating persons. The maximum value of velocity of air flow in the opposite direction should not be problematic for them.

3.7. Evacuation Model. As we can see from Figure 10, velocity of persons moving within the smoky space decreased even for the value of opacity 0.4 – (1/m). That value also changed based on position of persons in the tunnel, the tunnel did not get smoky immediately along the whole length, and the evacuating model should consider it.

The study of the smoke/speed correlation is currently based on two main datasets:

Set of experiments from Jin [1976]

Set of experiments from Frantzich and Nilsson [2003]

The first dataset (from Jin) collected data and they were used for providing correlation between the extinction coefficient and walking speeds, visibility levels, and cognitive abilities when exposed to smoke.

Jin used two types of different smoke. First was irritant smoke, which was produced by burning wood cribs. Second was non-irritant smoke, which was produced by burning kerosene. This experiment was performed in 20-meter long corridor that was filled with smoke corresponding to an early stage of fire. The experiment involved 17 women and 14 men, ranging from 20 to 51 years in age.

The second dataset (from Frantzich and Nilsson) is from more recent studies. This experiment was performed in tunnel for studying the influence of different visibility conditions on individual walking speeds.

Frantzich and Nilsson used artificial smoke and, for simulation irritation, acetic acid was used. This experiment was performed in 37-meter long tunnel tube. The experiment involved 46 people.

Cellular automaton (CA) model of traffic was extended for a number of persons inside vehicles; therefore, the evacuation model accepts an initial distribution of persons in the tunnel. Interconnection of the CA model of traffic and the evacuation model is shown in Figure 11.

To estimate speed of walking in the evacuation model, the following fuzzy inference system (FIS) was designed:

Density of persons (at the evacuation path): low ($<1.5\,(\text{person}/\text{m}^2)$), middle, and high ($>4\,(\text{person}/\text{m}^2)$)

Smokiness: low ($<(2/\text{m})$), middle, and high ($>(6/\text{m})$)

Illumination (the level of operational lighting): low (<25%), middle, and high (>70%)

In the first step, all membership functions were chosen as trapezoidal and linguistic variables were deduced expertly. It turned out it was possible to reach lower deviation from experimental data when tuning the shape of membership function by genetic algorithms. Comparison of various approaches is documented in [31, 32].

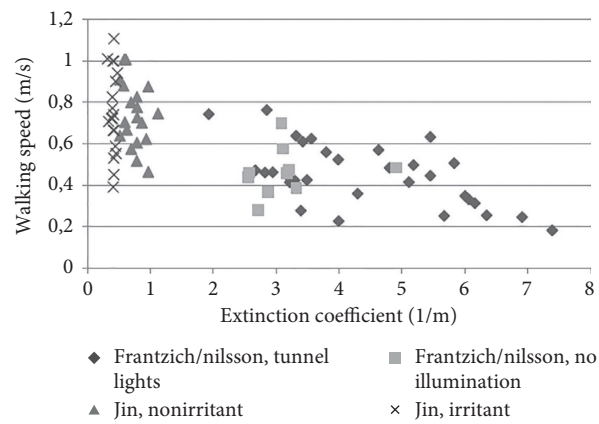


Figure 10: Walking speed of humans in smoky area, for switched-on and switched-off illumination [30].

To compare the evacuation model we decided to use data from [30] where detailed comparison of multiple evacuation tools is available. We repeated selected experiments with our evacuation model; comparison of input data for experiments is given in Table 1.

For all the scenarios, one-way traffic was applied, stopping at the emergency exit without switched-on ventilation. The scenarios A1.1, A1.2, and A2.1-A2.3 were analysed in [30]. The first two simulated a standard course of evacuation with its immediate initiation. Then, various walking speeds were tested. The average evacuation times for individual scenarios are given in Table 2 together with standard deviations (in parentheses).

4. Risk Analysis in TuSim

The TuSim extended for mathematical-physical models is a model suitable for simulation experiments with technological equipment based on scenario analysis. To evaluate scenarios, it is appropriate to compare ASET (Available Safe Egress Time) and RSET (Required Safe Egress Time), or mortality rates for individual scenarios. Meaning of times RSET and ASET is apparent from Figure 12.

Figure 13 shows a clearer time-spatial way of visualization of times RSET/ASET [35], where there is no problem to assess the situation at the particular place in the tunnel. On the left side of the picture, there is velocity of air flow indicated; on the right side of the picture, there is smoke propagation shown and the red lines represent escape paths to individual exits.

The way used to make simulation in TuSim is depicted in Figure 14. Setting of the simulation parameters is in the top left corner of the picture. In the top right part of the picture, there is the course of simulation for unstopped traffic and in the bottom right part for stopped traffic and evacuated persons.

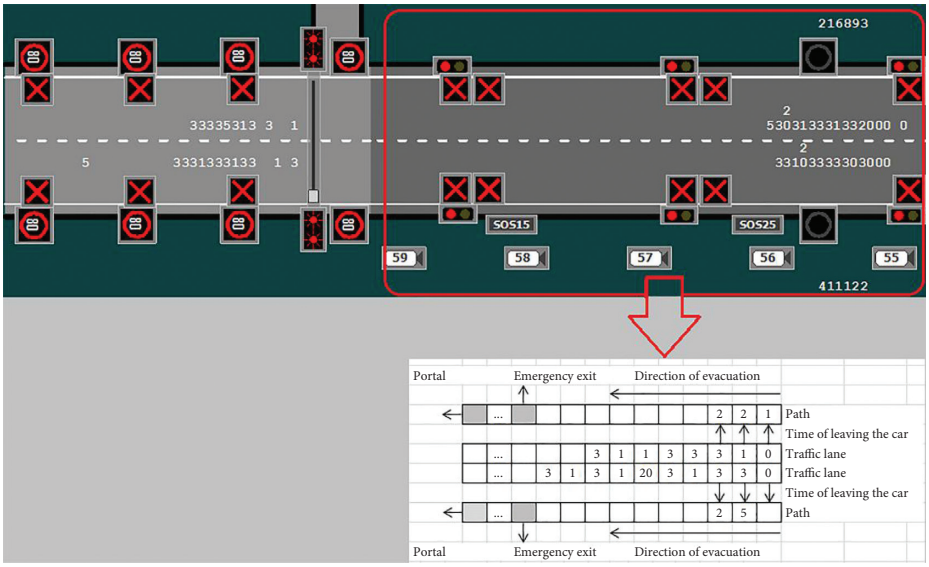


Figure 11: Interconnection of the evacuation with the CA model of traffic.

Table 1: Comparison of the evacuation models.

	Ronchi	TuSim
Tunnel length	670 m	675 m
Distance to emergency exit	390 m	390 m
Width of the path	2.5 m, 0.75 m	1.5 m, 1.5 m
Length of the vehicle	Car: 4.5 m + gap 1 m Truck: 10 m	7, 5 m with gaps
Number of persons in the vehicle	Cars: 2.5–5 trucks: 1-2	Cars: 3-6 trucks: 1-2
Number of persons in the tunnel	A1.1, A2.2 : 312 persons, A2.1, A2.1, A2.3 : 624 persons	Variable ~300 variable ~600

Table 2: Results of comparison of evacuation models.

Scenario	SFPE	FDS + EVACS	STEPS	Pathfinder	TuSim
A1.1	—	403	402	400	399 (16, 7)
A1.2	—	406	402	400	426 (11, 8)
A2.1	578	577 (18, 1)	583 (15, 4)	584 (12, 5)	583 (16, 6)
A2.2	508	545 (22, 6)	529 (19, 5)	535 (23, 1)	548 (23, 4)
A2.3	526	571 (6, 9)	535 (7, 9)	675 (17, 8)	562 (13, 9)

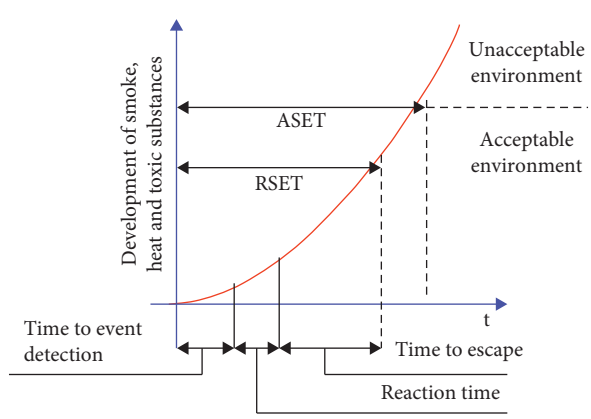


Figure 12: Meaning of times RSET and ASET [33].

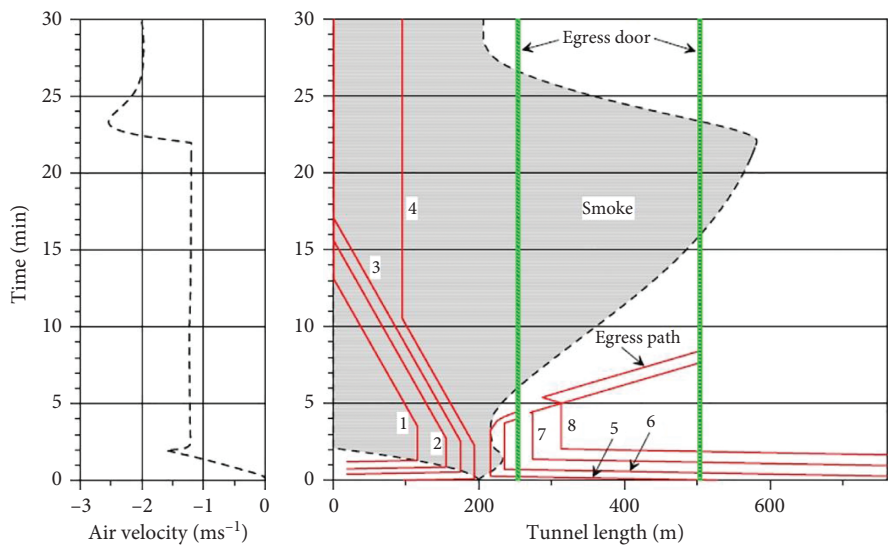


Figure 13: Graphical output of simulation of smoke propagation and escape paths [34].

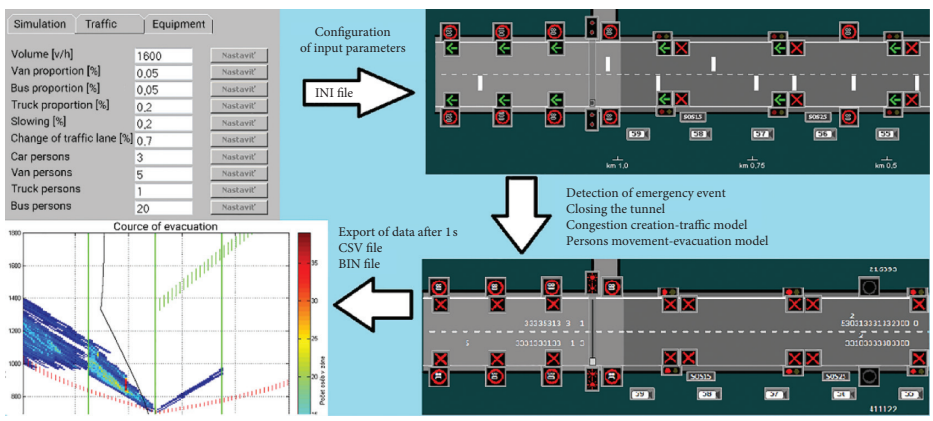


Figure 14: TuSim, simulation experiment.

5. Evaluation

We set the same conditions for scenarios:

Stopping of one-way traffic with the volume 1600 (veh/h) in the 300th minute of simulation

Response of the control system in the 600th minute of simulation

We were interested in the best case (switching-on of three pairs) and the worst case (switching-on of no fan).

To simulate the humans' decisions to evacuate, we used the rule of temperature increased to 45°C or smoke present at the place of human appearance. If none of the rules applied, we started evacuation in the 900th minute with time penalization according to distance from the place of fire. For each scenario, we showed time-spatial way of time visualization RSET/ASET, where red hatching indicates the value of air temperature 50 – 55°C.

Figure 15 shows that for fire of the truck velocity of air flow shortly decreased below the estimated value of critical velocity and backflow of smoke may occur for a short period of time. Increased temperature will cause earlier initiation of evacuation; temperature 50°C will reach emergency exit behind the fire place approximately in the 960th minute. If persons started evacuation in a wrong direction, in addition to smoke they will be endangered by heat as well.

Figure 16 shows the course of truck fire with switched-off ventilation where smoke backflow does not reach the emergency exit in front of the place of fire due to increased velocity of air flow to 1 (m/s). The persons complete evacuation by 1404 s. Temperature 50°C will reach the emergency exit behind the place of fire approximately by the 13th minute, emergency exit in front of the place of fire approximately by the 14th minute, and the whole evacuation of persons will run in environment with higher temperature. Higher temperature need not immediately cause inability of persons to evacuate, and smoke backflow below the ceiling of the tunnel need not immediately endanger persons. The smoke of the fire will start to mix with bottom layers of the air, even after partial cooling of the gas, and this effect can be very difficult considered in one-dimensional simulation models.

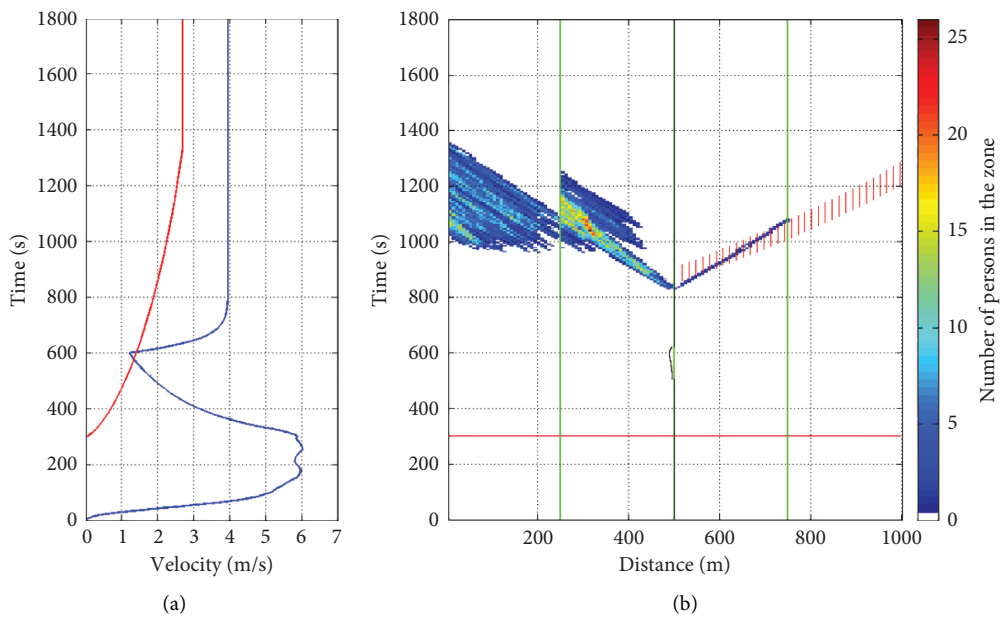


Figure 15: Course of evacuation for the truck fire with switched-on ventilation [36]. (a) Velocity of air flow. (b) Course of evaluation.

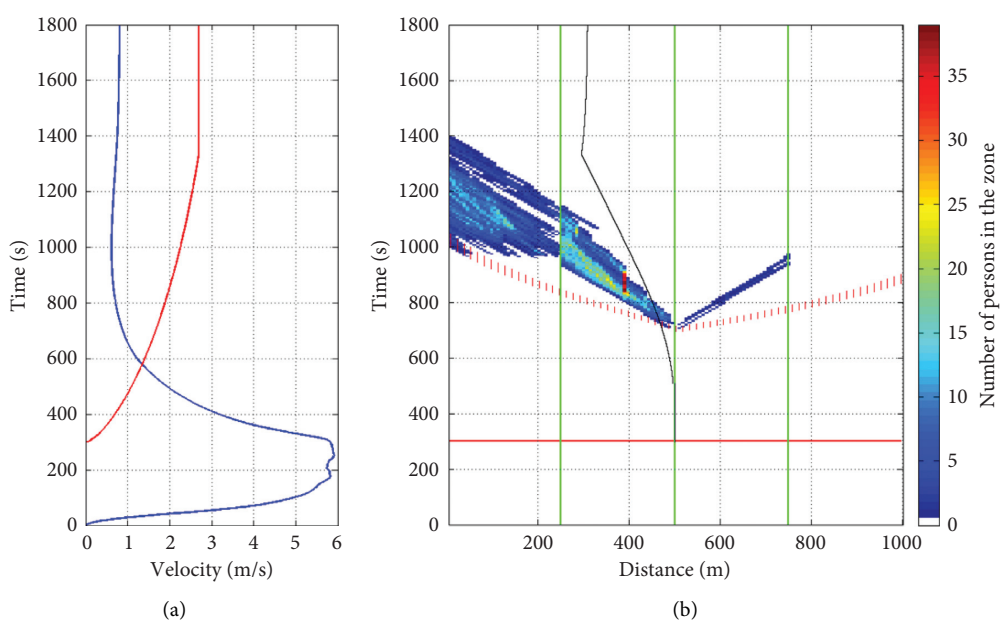


Figure 16: Course of evacuation for the truck fire with switched-off ventilation. (a) Velocity of air flow. (b) Course of evaluation.

Persson [34] utilized the one-dimensional model of the fire and used a fractional effective dose (FED) and fractional incapacitating dose (FID) for mortality calculation. For scenarios of vehicle fire, they got zero mortality, fatalities were caused only by the fire of dangerous goods, or fire with near to immediate increase to maximum power (explosive conflagration). For working ventilation, those findings are following our conclusions.

Three-dimensional simulations in the Horelica tunnel [37] revealed that temperature in the upper air mass in the height 1.8 m is not dangerous for humans in none of the analysed scenarios. The results are not valid for immediate surroundings of the fire, i.e., up to 10 m in front of the fire and behind it. In the direction of airflow, the temperature exceeded the value 50°C along the whole length of the tunnel tube for the fires 20 MW and 50 MW which corresponds to our findings for the lower velocities of airflow. For 50 MW fire, radiation above the level 2.5 kW/m^2 mostly endangered persons being the max. 25 m in the direction of airflow which corresponds to our calculated value of the distance 25.69 m for the same radiation and fire.

We can see that for the tunnel 1 km long we can keep the airflow in the direction of traffic only if fire detection and the

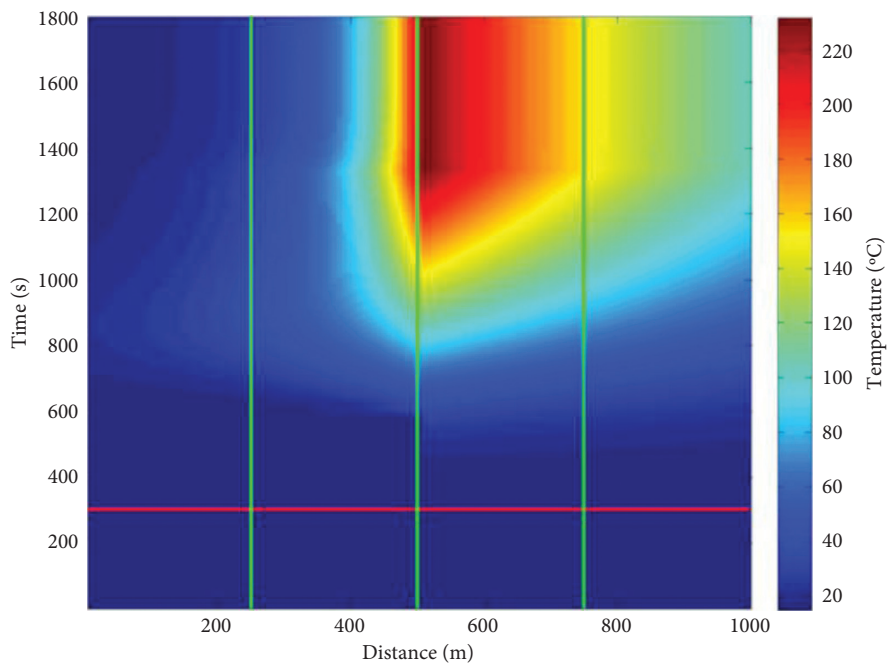


Figure 17: Development of temperature over space and time.

following start of the ventilation will occur within 5 minutes from the traffic stop. In the case of the correct function of fire detection and ventilation systems, without irrational behavior of the persons (evacuation in a wrong direction), without staying in vehicles, there will be no fatalities for one-way traffic.

In the case of two-way traffic, a part of persons will evacuate in a smoke environment with low visibility. Switching the ventilation off may be problematic in short tunnels due to a change in the airflow. Possible regulation to the given value of airflow, e.g., the particular values 0 or 1 m/s according to technical conditions, may be reachable with difficulties without the possibility to change the ventilation airflow (frequency inverters, fan blades angle of attack control, etc.).

6. Detection of Fire by the Vehicle

In the previous sections, we have proposed a method to simulate the physical conditions (opacity, temperature, air speed, carbon oxide concentration, etc.) inside the road tunnel in case of fire. As one could see in Figure 16, if the tunnel ventilation is not working, evacuating persons will be endangered by high temperature and smoke. Figure 17 shows temperature development over space and time inside the tunnel. First detectable rising of the temperature occurs at 600 s, but people will start to evacuate at 700 s. During those 100 s, the temperature will rise by approx. 20°C. The fire can be therefore detected in advance comparing the normal temperature inside the tunnel (computed by the TuSim without the event of fire) and the real measured temperature. If the temperature is above the maximal normal temperature and the vehicle is steady, there is a high possibility of the fire inside the tunnel and the autonomous vehicle has to issue an alert to its passengers prompting them to the nearest egress exit. Similar approach can utilize measurement of COx concentration indicating the presence of the smoke.

The alarm inside the vehicle should be issued when any of these conditions is met:

Vehicle is steady, and the temperature is above maximal normal temperature at given distance inside the tunnel

Vehicle is steady, and the concentration of COx is above the maximal normal level at given distance inside the tunnel

The maximal normal values for temperature and exhaust gasses concentration will be provided by TuSim for given parameters of the traffic and the tunnel. These values depend on the distance. The distance of the vehicle from the portal of the tunnel can be estimated by odometers inside the vehicle combined with other sensors used in autonomy navigation.

7. Conclusion

The article proposes a complex simulation method in order to predict the development of the physical properties (e.g., temperature, opacity) in case of fire inside the road tunnel. When evaluating an effect of the technological equipment in case of fire in the road tunnel, we must realize a detailed time-spatial analysis of the given event. The methods based on statistics are not able to consider effect of technologies when an event occurs. The correct functionality of detection and reaction of the tunnel control system is usually assumed. Further, the results from the statistics cannot be scaled linearly with the parameters of the tunnel. The only option is to perform a simulation experiment. An

advantageous option for simulation experiments is the use of PLC, as real control systems are built on the same platform. As a starting point of our work, we have used our TuSim tunnel simulator.

Once the possible scenarios (fire with different power) are computed, the predicted normal and critical behaviour of the tunnel along with the location of the emergency exits may be presented to the intelligent vehicles. The vehicles may detect the fire early considering changes in temperature and opacity. The proposed method is independent of the external systems in case of emergency (wireless communication inside the tunnel may fail in case of fire). Therefore, the passengers will be alerted even when the communication link with the tunnel is lost.

Data Availability

The data supporting the conclusions of this study are from the following: Miklóšik, I., Pokročilé metódy kvantifikácie bezpečnosti cestných tunelov. PhD. work, Faculty of Electrical Engineering and Information Technology, University of Žilina, 2016, http://kris.uniza.sk/images/dokumenty/EU_CEx_IDS.pdf http://kris.uniza.sk/veda-vyskum-prax/grant-ulohy.

Conflicts of Interest

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

Acknowledgments

This work was supported by Slovak Research and Development Agency under APVV-17-0014.

References

[1] B. Luin, S. Petelin, and P. Vidmar, "Interactive model OF a road tunnel during," in *Proceedings of the 14th International Conference on Transport Science - ICTS 2011*, p. 10, Portorož, Slovenia, 2011.

[2] Federal Highway Administration of Transport. Federal Highway Administration of Transport. https://www.fhwa.dot.gov/.

[3] EQUA Simulation, "IDA road tunnel ventilation 3.0," 2017, http://www.ilg-engineering.ch/Tunnelsicherheit/IDA_Tunnel_brochure.pdf.

[4] Časopis Českého Tunelárského Komitétu a Slovenskej Tunelárskej Asociácie ITA/AITES. http://www.ita-aites.cz/files/tunel/komplet/tunel_4_07.pdf.

[5] I. R. Riess, P. Altenburger, and P. Sahlin, "On the design and control of complex tunnel ventilation," in *Proceedings of the 12th Int. Symp. Aerodynamics and Ventilation of Vehicle Tunnels*, p. 11, Portoroz. Portoroz, 2006.

[6] GE Fanuc Automation, "CIMPLICITY HMI plant edition," 2017, http://platforma.astor.com.pl/files/getfile/id/4664.

[7] GE Fanuc Automation, "CIMPLICITY HMI plant edition - CimEdit operation manual," 2017, http://platforma.astor.com.pl/files/getfile/id/4675.

[8] LIBNODAVE, "Exchange data with Siemens PLCs," 2017, http://libnodave.sourceforge.net/.

[9] Mathworks, "Matlab help," 2017, http://www.mathworks.com/help/matlab/index.html.

[10] I. Miklóšik, J. Spalek, M. Hruboš, and P. Přibyl, "Cellular automaton traffic flow model with vehicle type and number of persons," in *Archives of Transport System Telematics*p. 5, PSTT, Wrolaw, Poland, 2015.

[11] I. Miklóšik and J. Spalek, "Extension of the tunnel simulator with the traffic flow model," in *Telematics – Support for Transport*, J. Mikulski, Ed., Springer, Ustroń, Poland, pp. 156–165, 2014.

[12] H. J. Fernando, *Handbook of Environmental Fluid Dynamics, Volume Two: Systems, Pollution, Modeling, and Measurements*, CRC Press, vol. 12, p. 587, Boca Raton, FL, USA, 2012.

[13] D. Wei and W. Ge, "Research on one bio-inspired jumping locomotion robot for search and rescue," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, p. 168, 2014.

[14] L. Kurka, L. Ferkl, O. Sládek, and J. Porízek, "Simulation of traffic, ventilation and exhaust in a complex road tunnel," in *Proceedings of the IFAC Volumes (IFAC-PapersOnline); 2005*, pp. 60–65, IFAC, Prague, Czech Republic, July 2005.

[15] Comité technique AIPCR, "Exploitation des tunnels routiers/PIARC Technical Committee C.4 Road Tunnel Operation. Road tunnels: vehicle emissions and air demand for ventilation," *Environment/Road Tunnel Operations*, vol. 87, 2012.

[16] P. Lunardi, G. Cassani, M. Gatti, G. Lodigiani, M. Frankovský, and M. Fulvio, "The ADECO-RS approach and the full-face excavation," *Tunely a podzemné stavby*, vol. 11, p. 12, 2015.

[17] Y. Z. Li, C. G. Fan, H. Ingason, A. Lönnermark, and J. Ji, "Effect of cross section and ventilation on heat release rates in tunnel fires," *Tunnelling and Underground Space Technology*, vol. 51, pp. 414–423, 2016.

[18] M. Pavelka and P. Přibyl, *Simulace Pohybu Vzduchu a Škodlivin V Tunelu–Matematický Model*, ČVUT v Praze Fakulta Dopravní, Konviktská, Czechia, 2006.

[19] M. dopravy and S. R. výstavby a regionálneho rozvoja, "Analýza rizík pre slovenské cestné tunely," 2017, http://www.ssc.sk/files/documents/technicke-predpisy/tp2011/tp_02_2011.pdf.

[20] National Cooperative Highway Research Program (NCHRP), *Synthesis 415: Design Fires in Road Tunnels*, National Cooperative Highway Research Program (NCHRP), Washington, DC, 2011.

[21] D. Gola, *Simulace Aerodynamického Chování*, https://dspace.cvut.cz/bitstream/handle/10467/61855/F3-BP-2015-Gola-Daniel-priloha-bakalarska_prace.pdf, p. 59, České vysoké učení technické v Praze, Prague, Czech Republic, 2015, .

[22] I. Riess and M. Bettelini, "The prediction of smoke propagation due to tunnel fires," in *Proceedings of the ITC Conference Tunnel Fires and Escape from Tunnels*, p. 18, Lyon, France, 1999.

[23] NIST - National Institute of Standards and Technology, "CFAST, Fire Growth and Smoke Transport Modeling," 2017, https://www.nist.gov/el/fire-research-division-73300/product-services/consolidated-fire-and-smoke-transport-model-cfast.

[24] S. Tavelli, R. Rota, and M. Derudi, "A critical comparison between CFD and zone models for the consequence analysis of fires in congested environments," *Chemical Engineering Transactions*, vol. 36, p. 6, 2014.

[25] Bundesministeriums für Verkehr, Innovation und Technologie . Betrachtung der Wärmefreisetzung im Brandfall. https://www.bmvit.gv.at/service/publikationen/verkehr/strasse/downloads/tunnel_laengslueftung.pdf.

[26] I. Riess and M. Bettelini, "Smoke extraction in tunnels with considerable slope," in *Proceedings of the 4th International*

Conference Safety in Road and Rail Tunnels, pp. 503–512, Madrid, Spain, 2001.

[27] I. Miklóšik, P. Kello, and J. Spalek, "Fiber laser fire detection in the tunnel simulator," in *Proceedings of the 2016 ELEKTRO*, pp. 429–434, IEEE, Strbske Pleso, Slovakia, 2016.

[28] S. A. T. RA. Kvantitativní Analýza Přepravy Nebezpečných Nákladů Silničním Tunelem Sitina V Bratislavě. http://www.ita-aites.cz/files/tunel/2007/3/tunel-0703-5.pdf.

[29] M. Banjac, "Numerical study of smoke flow control in tunnel fires using ventilation systems," *FME Transactions*, vol. 36, p. 2008, 2008.

[30] E. Ronchi, "Evacuation modelling in road tunnel fires," 2008, http://lup.lub.lu.se/search/ws/files/5519346/4001478.pdf.

[31] M. Gregor, I. Miklóšik, and J. Spalek, "Automatic tuning of a fuzzy meta-model for evacuation speed estimation," in *Proceedings of the 2016 Cybernetics & Informatics (K&I)*, pp. 1–6, Levoca, Slovakia, 2016.

[32] P. Matis and J. Spalek, "ATP journal plus 2/2013 - fuzzy model doby reakcie osôb v cestnom tuneli pri vzniku mimoriadnej udalosti," 2017, http://www.atpjournal.sk/buxus/docs/casopisy_cele/ATP_PLUS_2_2013_zmensene.pdf.

[33] I. Riess and R. Brandt, "A one-dimensional egress model for risk analysis," in *Proceedings of the Symposium Tunnel Safety and Ventilation*, vol. 5, pp. 165–172, Graz, Austria, 2010.

[34] A. Osvald, V. Mózer, and J. Svetlík, *Požiarna Bezpečnosť Cestných Tunelov*, p. 140, EDIS, Žilina, Slovakia, 2014.

[35] M. Persson, "Quantitative risk analysis procedure for the fire evacuation of a road tunnel," 2017, https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=1688790&fileOId=1765306.

[36] P. Kello, I. Miklóšik, J. Spalek, and T. Tichý, *The Comparison of Selected Fire Scenarios with TuSim*, ELEKTRO, Mikulov. Czech Republic, pp. 1–4, 2018.

[37] J. Hrbcek and V. Šimak, "Implementation of multi-dimensional model predictive control for critical process with stochastic behavior," *Advanced Model Predictive Control*, pp. 109–124, 2011.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 8878011, 18 pages
https://doi.org/10.1155/2021/8878011

Review Article

A Review of Traffic Congestion Prediction Using Artificial Intelligence

Mahmuda Akhtar and Sara Moridpour

Department of Civil and Infrastructure Engineering, RMIT University, Melbourne, VIC 3000, Australia

Correspondence should be addressed to Mahmuda Akhtar; s3799862@student.rmit.edu.au

Received 8 August 2020; Revised 7 January 2021; Accepted 18 January 2021; Published 30 January 2021

Academic Editor: Michael Bazant

In recent years, traffic congestion prediction has led to a growing research area, especially of machine learning of artificial intelligence (AI). With the introduction of big data by stationary sensors or probe vehicle data and the development of new AI models in the last few decades, this research area has expanded extensively. Traffic congestion prediction, especially short-term traffic congestion prediction is made by evaluating different traffic parameters. Most of the researches focus on historical data in forecasting traffic congestion. However, a few articles made real-time traffic congestion prediction. This paper systematically summarises the existing research conducted by applying the various methodologies of AI, notably different machine learning models. The paper accumulates the models under respective branches of AI, and the strength and weaknesses of the models are summarised.

1. Introduction

Artificial intelligence (AI) is the most important branch of computer science in this era of big data. AI was born 50 years ago and came a long way, making encouraging progress, especially in machine learning, data mining, computer vision, expert systems, natural language processing, robotics, and related applications [1]. Machine learning is the most popular branch of AI. Other classes of AI include probabilistic models, deep learning, artificial neural network systems, and game theory. These classes are developed and applied in a wide range of sectors. Recently, it has been the leading research area in transportation engineering, especially in traffic congestion prediction.

Traffic congestion has a direct and indirect impact on a country's economy and its dwellers' health. According to Ali et al. [2], traffic congestion causes Pak Rs. 1 million every day in terms of opportunity cost and fuel consumption due to traffic congestion. Traffic congestion affects on individual level as well. Time loss, especially during peak hours, mental stress, and the added pollution to the global warming are also some important factors caused due to traffic congestion.

Ensuring economic growth and the road users' comfort are the two requirements for the development of a country, which is impossible without smooth traffic flow. With the development in the transportation sector by collecting traffic information, authorities are putting more attention on traffic congestion monitoring. Traffic congestion prediction provides the authorities with the required time to plan in the allocation of resources to make the journey smooth for travellers. Traffic congestion prediction problem discussed in this paper can be defined as an estimation of parameters related to traffic congestion into the short-term future, e.g., 15 minutes to a few hours by applying different AI methodologies by using collected traffic data. There are usually five parameters to evaluate, including traffic volume, traffic density, occupancy, traffic congestion index, and travel time while monitoring and predicting traffic congestions. Depending on the nature of the collected data, a variety of AI approaches are applied to evaluate the congestion parameters. This article systematically discusses the models and their advantage and disadvantages. The primary motivation of this review is to gather the articles focusing solely on traffic congestion prediction models. The keywords used in

the search process included "traffic congestion prediction" OR "traffic congestion estimation" OR "congestion prediction modelling" OR "prediction of traffic congestion" OR "road congestion forecast" OR "traffic congestion forecast." For efficient screening, research paper search was done according to year using search engines like Scopus, Google Scholar, and Science Direct. After collecting all the peer-reviewed journal and conference papers written in the English language, 48 articles were found for review. Any studies focusing on the cause of traffic congestion, traffic congestion control, traffic congestion impact, traffic congestion propagation, traffic congestion prevention, etc. were excluded from this manuscript.

A general layout of the prediction approaches is provided in Section 2. The data collection sources and congestion forecasting models are explained in Sections 3–6 and they provide the overall discussion and concluding remarks.

2. General Layout

Traffic congestion forecasting has two basic steps of data collection and prediction model development. Every step of the methodology is important and may affect the results if not done correctly. After data collection, data processing plays a vital role to prepare the training and testing datasets. Case area differs for different research. After developing the model, it is validated with other base models and ground true results. Figure 1 shows the general components of traffic congestion prediction studies. These branches were further divided into more specific sub-branches and are discussed in the following sections.

3. Data Source

Traffic datasets used in different studies can be mainly divided into two classes, including stationary and probe data. Stationary data can be further divided into sensor data and fixed cameras. On the other hand, probe data that were used in the studies were GPS data mounted on vehicles.

Stationary sensors continuously capture spatiotemporal data of traffic. However, sensor operation may interrupt anytime. Authorities should always consider this temporary failure of the sensor while planning by using this data. The advantage of the sensor data is that there is no confusion on the location of the vehicles. The most used dataset was Performance Measurement System (PeMS) that collects highway data across all major metropolitan areas of the State of California of traffic flow, sensor occupancy, and travel speed in real-time. Most of the studies used dataset from the I-5 highway, in San Diego, California, every 5 minutes [3–6]. Other systems included the Genetec blufaxcloud travel-time system engine (GBTTSE) [7] and the Topologically Integrated Geographic Encoding and Referencing (TIGER) line graph [8].

On the other hand, probe data has the advantage of covering the entire road network. A network consists of different structured roads. Therefore, studies, especially those that considered the network wide area, used probe data. The most used dataset was GPS data collecting every second from approximately 20000 taxies of Beijing, China. Data included the taxi number, the latitude-longitude of the vehicle, timestamp when sampling, and whether there was a passenger or not. Data updating frequency of this dataset varies from 10 s to 5 min according to the quality of GPS device [4, 5, 9]. Other probe data included low-frequency Probe Vehicle Data (PVD) [10] and bus GPS data [11, 12]. However, sometimes probe data show significant fluctuation. Besides, map matching is usually a must for probe data. But data can minimize this limitation. Probe data collected from one city cannot be used directly for modelling other city networks. This is because the data collected from Beijing, China, includes latitude-longitude of the vehicle, which is unique. However, a generalised model using probe data can be generated for different cities.

Other data sources, e.g., data from tolling system and data provided by transportation authority, will add more reliable data as the sources are dependable. However, a lot of the times, study area needs to be adjusted as in most cases, tolled road information is not available. Tracking cellular phone movements without privacy breach can also be a source of data. However, the heterogeneity of the vehicle distribution will be hard to determine from this dataset, if not impossible. Besides, due to pedestrian or cyclists travelling through the sidewalk, there might be many outliers in the dataset if modelling is done for a road network. Data collected from a questionnaire to the general public/drivers may provide a misleading result [13].

3.1. Clustering Algorithms. Some studies use clustering the acquired data before applying the main congestion models of prediction. This hybrid modelling technique is applied to fine-tune the input values and to use them in the training phase. Figure 2 shows the commonly used AI clustering models in this field of research. The models are described briefly in this section.

Fuzzy C-Means (FCM) is a popular nondeterministic clustering technique in data mining. In traffic engineering researches, traffic pattern recognition plays an important role. Besides, these studies often face the limitation of missing or incomplete data. To deal with these constraints, FCM has become a commonly applied clustering technique. The advantage of this approach is, unlike original C-means clustering methods, it can overcome the issue of getting trapped in the local optimum [14]. However, FCM requires setting a predefined cluster number, which is not always possible while dealing with massive data without any prior knowledge of the data dimension. Besides, this model becomes computationally expensive with data size increment. Different studies have applied FCM successfully by improving its limitations. Some studies changed the fuzzy index value for each FCM algorithm execution [15], some calculated the Davies-Bouldin (DB) index [10], while others applied the K-means clustering algorithm [16, 17].

K-means clustering is an effective and relatively flexible algorithm while dealing with large datasets. It is a popular unsupervised machine learning algorithm. Depending on the features, cluster number varied from two [18] to 50

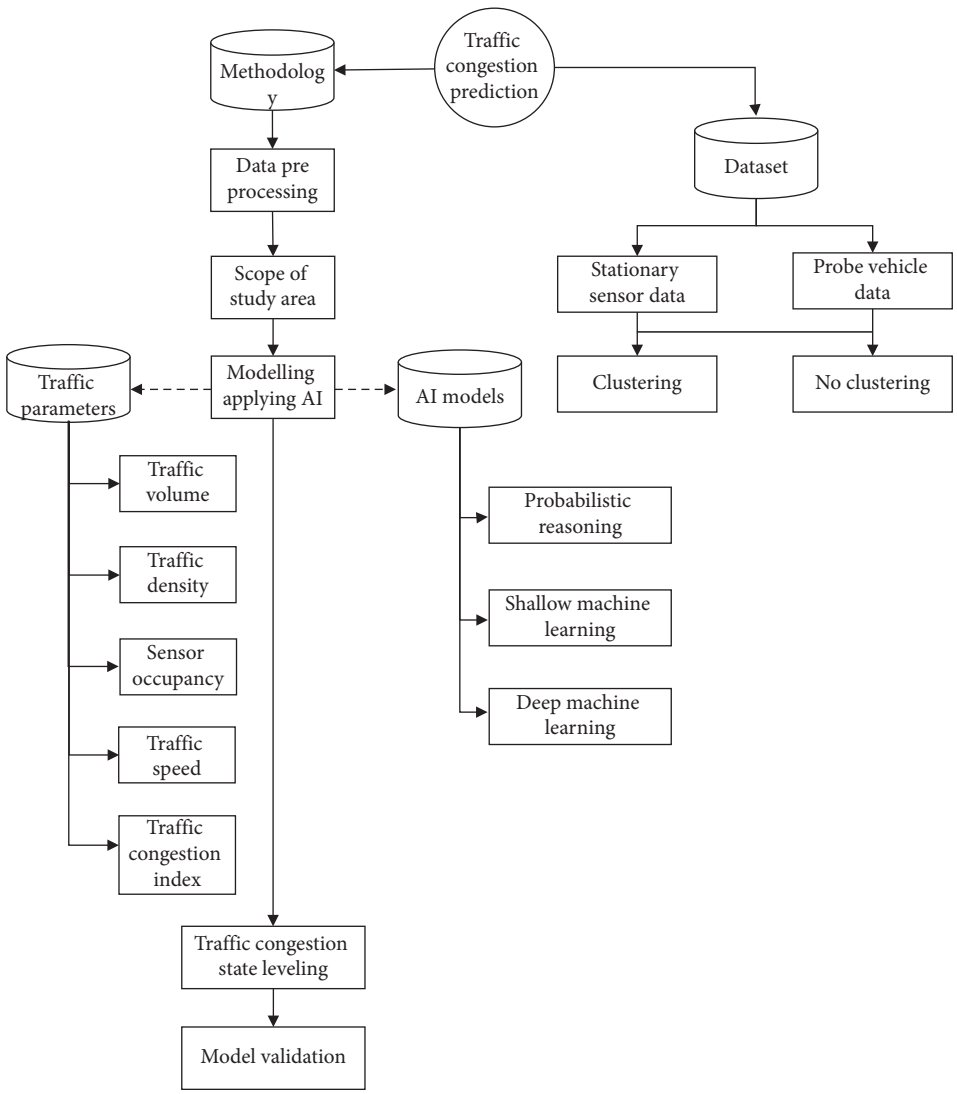


Figure 1: The layout of traffic congestion prediction system.

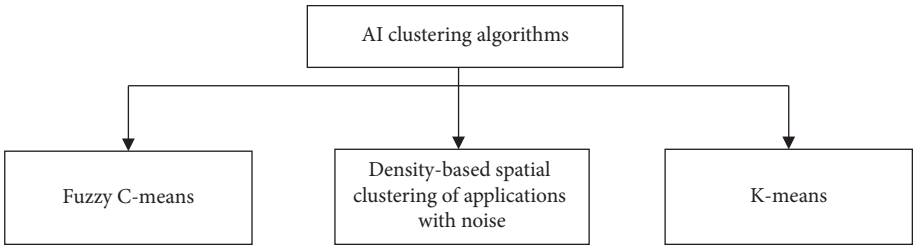


Figure 2: Commonly used AI clustering algorithms.

[19–21]. Like FCM, K-means clustering requires a predefined cluster number and selecting K original cluster centres. GAP [22] and WEKA toolbox [23] were used to define the value. For large datasets, as the sample distribution is unknown in the beginning, it is not always possible to fulfil these two requirements. A few studies used adaptive K-means clustering overcoming the limitations and exploited the pattern using principal component analysis (PCA) [24, 25].

DBSCAN is more of a general clustering application in machine learning and data mining. This method overcomes the limitation of FCM of predefining the cluster number. It

can automatically generate arbitrary cluster shapes surrounded by clusters of different characteristics and can easily recognise outlier. However, it requires two parameters to preset. A suitable parameter determination method, e.g., trial and error method [8] and human judgement [26] makes the model computationally expensive and requires a clear understanding of the dataset.

From the above discussion, it is concluded that only 16 out of 48 studies have done clustering before applying prediction models. Several time-series models and shallow machine learning (SML) algorithms have used clustering approach. However, deep learning algorithms can process input data on different layers of the model, thus may not need clustering beforehand.

4. Applied Methodology

Traffic flow is a complex amalgamation of heterogenous traffic fleet. Thus, traffic pattern prediction modelling could be an easy and efficient congestion prediction approach. However, depending on the data characteristics and quality, different classes of AI are applied in various studies. Figure 3 shows the main branches—probabilistic reasoning and machine learning (ML). Machine learning comprised of both shallow and deep learning algorithms. However, with the progress of this article, these sections were subdivided into detailed algorithms.

To generalise traffic congestion forecasting studies using different models is not straight forward. The common factors of all the articles include the study area, data collection horizon, predicted parameter, prediction intervals, and validation procedure. Most of the articles took studied corridor segment as the study area [5, 27–30]. Other study areas included the traffic network [31, 32], ring road [9], and arterial road [33]. Data collection horizon varied from 2 years [34] to less than a day [35] in the studies. Congestion estimation is done predicting traffic flow parameters, e.g., traffic speed [4], density, speed [5], and congestion index [31], to mention a few. The Congestion Index (CI) approach is suitable to monitor the congestion level continuously in a spatiotemporal dimension. Studies those compared their results with the ground truth value or with other models used mean absolute error (MAE) (equation (1)), symmetric mean absolute percentage error (sMAPE) (equation (1)), MAPE, root-mean-squared error (RMSE) (equation (3)), false positive rate (FPR) (equation (4)), and detection rate (DR) (equation (5)). Many studies used SUMO to validate their models:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \overline{Y}_i\right|, \tag{1}$$

$$\mathrm{sMAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\overline{Y}_i - Y_i\right|}{\left(\left|\overline{Y}_i\right| + \left|Y_i\right|\right)/2}.\,100, \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(\overline{Y}_i - Y_i\right)^2}{n}}, \tag{3}$$

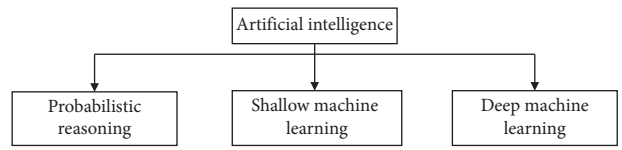


Figure 3: Branches of artificial intelligence in this article.

where $\overline{Y}$ = original value, Y_i = predicted value, and n = number of instances.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}, \tag{4}$$

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}}, \tag{5}$$

where FP, TN, FN, and TP represent the false positive, true negative, false negative, and true positive, respectively.

The rest of this section will discuss the methodology the authors have applied in the studies.

4.1. Probabilistic Reasoning. Probabilistic reasoning is a significant section of AI. It is applied to deal with the field of uncertain knowledge and reasoning. A variety of these algorithms are commonly used in traffic congestion prediction studies. The studies discussed hereunder probabilistic reasoning is shown in Figure 4.

4.1.1. Fuzzy Logic. Zadeh is a commonly applied model in dynamic traffic congestion prediction as it allows vagueness instead of binary outcomes. In this method, several membership functions are developed those represent the degree of truth. With the vastness with time, traffic data are becoming complex and nonlinear. Due to its ability to deal with uncertainty in the dataset, fuzzy logic has become popular in traffic congestion prediction studies.

A fuzzy system comprises of several fuzzy sets, which is built of membership functions. There are usually three codification shapes to choose for the membership functions (MFs) of input: triangular, trapezoidal, and Gauss function. The fuzzy rule-based system (FRBS) is the most common fuzzy logic system in traffic engineering research. It consists of several IF-THEN rules that logically relate the input variables with output. It can effectively deal with the complexity resulting from real-world traffic situations by representing them in simple rules. These rules combine the relations among different traffic states to detect the resulting traffic condition [36]. However, with the growth in data complexity, the total number of rules also grows, lessening the accuracy of the whole system, thus making it computationally expensive. To better manage this problem, two types of fuzzy logic controls are applied. In hierarchical control (HFRBS), according to the significance, the input variables are ordered and MFs are employed. Figure 5 shows a simple HFRBS structure. MFs are optimized by applying different algorithms, e.g., genetic algorithm (GA) [30], hybrid genetic algorithm (GA), and cross-entropy (CE)

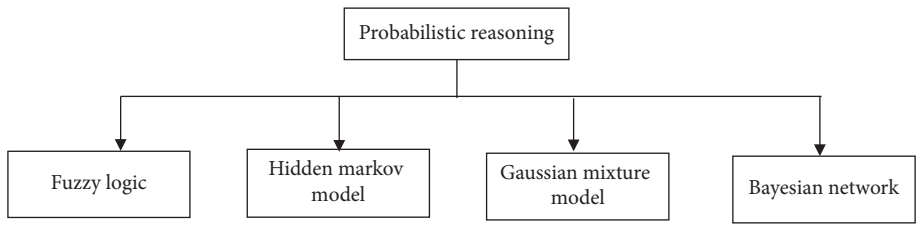


Figure 4: Subdivision of probabilistic reasoning models.

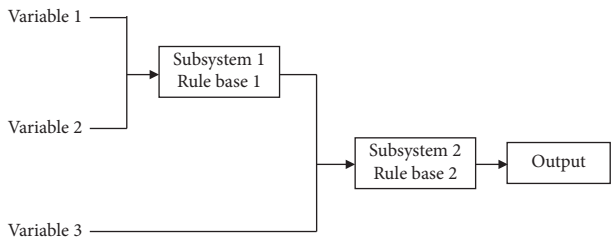


Figure 5: A simple structure of HFRBS.

[28, 37] compared the performance of evolutionary crisp rule learning (ECRL) and evolutionary fuzzy rule learning (EFRL) for road traffic congestion prediction. It was seen that ECRL models outperformed EFRL in terms of averaged accuracy and no of rules but was computationally expensive.

The Takagi-Sugeno-Kang (TSK) (FRBS) model is one of the simple fuzzy models due to its mathematical treatability. A weighted average computes the output of this model. Another simple FRBS model is Mamdani-type model. The output of this model is a fuzzy set which needs defuzzification, which is time-consuming. Due to its good interpretability, it can improve the accuracy of fuzzy linguistic models. Cao and Wang [3] applied this model to show the congestion severity change among road grades. A few studies used this method to fuse heterogenous parameters [7, 13]. The TSK model works on improving the interpretability of an accurate fuzzy model. TSK is applied for its fast calculation characteristics [37].

The fuzzy comprehensive evaluation (FCE) uses the principle of fuzzy transformation and maximum membership degree. This model consists of several layers, which is a useful objective evaluation method, assessing all relevant factors. The number of layers depends on the objective complicacy and the number of factors. Kong et al. [4] and Yang et al. [5] applied FCE in which the weights and the fuzzy matrix of multi-indexes were adapted according to the traffic flow to estimate traffic congestion state. Adaptive control adjusts weight coefficient based on judgement matrix. Certain weights are assigned to calculate the membership degree of the parameters [35].

Other than GA and PSO, Ant Colony Optimization (ACO) algorithm was also introduced by Daissaoui et al. [38] in fuzzy logic system. They provided the theory for a smart city, where each vehicle GPS data was taken as a pheromone, consistent with the concept of ACO. The objective was to predict traffic congestion one minute ahead from the information (pheromone) provided by past cars. However, the article does not give any result on support to the model.

As discussed before, with the development of optimisation algorithms, optimisation of the fuzzy logic system's membership functions is becoming diverse. With time, the simplest form of FRBS-TSK has become popular due to its good interpretability. Some other sectors of transportation where fuzzy logic models are popular include traffic light/signal control [39, 40], traffic flow prediction (Zhang and Ye [41]), traffic accident prediction [42], and modified fuzzy logic for freeway travel time estimation (Zhang and Ge [43]).

The fuzzy logic system is the only probabilistic reasoning model that can have an outcome of more than congested/noncongested state of the traffic state. This is one of the main advantages that has made this methodology popular. However, no study has provided any reasonable logic on selecting the membership function, which is a significant limitation of fuzzy logic models.

4.1.2. Hidden Markov Model. The hidden Markov model (HMM) is a combination of stochastic characteristics of Markov process and discrete characteristics of Markov chains. It is a stochastic, time-series event recognition technique. Some studies have applied Markov chain model for traffic pattern recognition during congestion prediction [21, 25, 44]. Pearson correlation coefficient (PCC) is commonly applied among the parameters during pattern construction. Zaki et al. [32] applied HMM to select the appropriate prediction model from several models they developed applying the adaptive neurofuzzy inference system (ANFIS). They obtained optimal state transition by four processing steps: initialization, recursion, termination, and backtracking. The last step analysed the previous step to determine the probability of the current state by using the Viterbi algorithm. Based on the log-likelihood of the initial model parameter, defined by expectation maximization (EM) algorithm, of HMM with the traffic pattern, a suitable congestion model was selected for prediction. Mishra et al. [23] applied the discretised multiple symbol HMM (MS-HMM) prediction model named future state prediction (FSP). They evaluated model adaptability for different road segments. A label was generated containing hidden states of MS-HMM, and the output was used for FSP to result in the next hidden state label.

In traffic engineering, especially while utilising probe vehicle data, HMM is very useful in map-matching. Sun et al. [45] applied HMM for mapping the trajectory of observed GPS points in nearby roads. These candidate points were taken as hidden states of HMM. The candidate points closer to the observation point had higher observation probability.

Transition probability of two adjacent candidates was also considered to avoid the misleading results generated from abrupt traffic situations.

HMM shows accuracy in selecting a traffic pattern or a traffic point. It has the advantage that it can deal with the data with outliers. However, points with a short sampling interval seem to be matched well, and long intervals and higher similar probe data decreased the model accuracy. Studies have found a significant mismatch for long sampling interval dataset and similar road networks.

The GPS tracking system has been widely developed in this era of the satellite. Thus, making HMM modelling is currently more relevant for map matching. Other sectors of transport where HMM is applied include traffic prediction [46], modified HMM for speed prediction [47], and traffic flow state transition [48], etc.

4.1.3. Gaussian Distribution. Gaussian processes have proven to be a successful tool for regression problems. Formally, a Gaussian process is a collection of random variables, any finite number of which obeys a joint Gaussian prior distribution. For regression, the function to be estimated is assumed to be generated by an infinite-dimensional Gaussian distribution, and the observed outputs are contaminated by additive Gaussian noise.

Yang [29] applied Gaussian distribution for traffic congestion prediction in their study. This study was divided into three parts. First, the sensor ranking was done according to the volume quality by applying p test. In the second part of the study, the congestion occurring probability was determined from a statistics-based method. In the learning phase of this part, two Gaussian probability models were developed from two datasets for every point of interest. In the decision phase, on which model the input traffic volume value fitted was evaluated, and a prediction score presenting congestion state was determined from the ratio of two models. Finally, the probability of congestion occurring at the point of interest was found by combining and sorting the prediction score from all the ranked sensors. Zhu et al. [49] also presented the probability of traffic state distribution. Selection of mean and variance parameters of Gaussian distribution is an important step. In this study, the EM algorithm was applied for this purpose. The first step generated the log-likelihood expectation for the parameters, whereas the last step maximised it. Sun et al. [45] approximated the error in GPS location in the road with Gaussian Distribution, taking mean 0. The error was calculated from the actual GPS point, matching point on the road section, and standard deviation of GPS measurement error.

From the abovementioned studies, it is seen that the Gaussian distribution model has a useful application in reducing feature numbers without compromising the quality of the prediction results or for location error estimation while using GPS data. Gaussian distribution is also applied in traffic volume prediction [50], traffic safety [51], and traffic speed distribution variability [52].

4.1.4. Bayesian Network. A Bayesian network (BN), also known as a causal model, is a directed graphical model for representing conditional independencies between a set of random variables. It is a combination of probability theory and graph theory and provides a natural tool for dealing with two problems that occur through applied mathematics and engineering—uncertainty and complexity [53].

Asencio-Cortés et al. [54] applied an ensemble of seven machine learning algorithms to compute the traffic congestion prediction. This methodology was developed as a binary classification problem applying the HIOCC algorithm. Machine learning algorithms applied in this study were K-nearest neighbour (K-NN), C4.5 decision trees (C4.5), artificial neural network (ANN) of backpropagation technique, stochastic gradient descent optimisation (SGD), fuzzy unordered rule induction algorithm (FURIA), Bayesian network (BN), and support vector machine (SVM). Three of these algorithms (C4.5, FURIA, and BN) can produce interpretable models of viewable knowledge. A set of ensembled learning algorithms were applied to improve the results found from these prediction models. The ensemble algorithm group included bagging, boosting (AdaBoost M1), stacking, and Probability Threshold Selector (PTS). The authors found a significant improvement in Precision for BN after applying ensemble algorithms. On the other hand, Kim and Wang [34] applied BN to determine the factors that affect congestion initialization on different road sections. The developed model of this study gave a framework to assess different scenario ranking and prioritizing.

Bayesian network is seen to perform better with ensembled algorithms or while modified, e.g., other transport sectors of traffic flow prediction [55] and parameter estimation at signalised intersection [56, 57].

4.1.5. Others. Other than the models mentioned above, the Kalman Filter (KF) is also a popular probabilistic algorithm. With the increment of available data, data fusion methods are becoming popular. The fusion of historical and real-time traffic data can achieve a higher level of traffic congestion prediction accuracy. In this regard, KF is commonly applied. Extended KF (EKF) is an extension of KF, which can be used to stochastically filter the nonlinear noises to improve the mean and covariance of an estimated state. Therefore, after data fusion, it updated the estimated covariance error by removing outliers [7].

Wen et al. [8] applied GA in traffic congestion prediction from spatiotemporal traffic environment. Temporal association rules were extracted from the traffic environment applying GA-based temporal association rules (GATARs). Their proposed Hybrid Temporal Association Rules Mining method (HTARM) included DBSCAN and GATAR methods. The DBSCAN application method was discussed previously in this article. While encoding using GATAR, road section number and congestion level were included in the chromosome. The decoding was done to obtain temporal association rules and was sorted according to confidence and support value in the rule pool. For both simulated and real-

world scenarios, the proposed HTARM method outperformed GATAR in terms of extracting temporal association rules and prediction accuracy. However, the cluster number difference showed a big difference in the two scenarios. Besides, with the increment of road network complexity, the prediction accuracy decreased.

Table 1 summarises the methodologies and different parameters used in various studies we have discussed so far.

4.2. Shallow Machine Learning. Shallow machine learning (SML) algorithms include traditional and simple ML algorithms. These algorithms usually consist of a few, many times, one hidden layer. SML algorithms cannot extract features from the input, and features need to be defined beforehand. Model training can only be done after feature extraction. SML algorithms and their application in traffic congestion studies are discussed in this section and shown in Figure 6.

4.2.1. Artificial Neural Network. Artificial neural network (ANN) was developed, mimicking the function of the human brain to solve different nonlinear problems. It is a first-order mathematical or computational model that consists of a set of interconnected processors or neurons. Figure 7 shows a simple ANN structure. Due to its easy implementation and efficient forecasting ability, ANN has become popular in the field of traffic congestion prediction research. Hopfield network, feedforward network, and backpropagation are the examples of ANN. Feedforward neural network (FNN) is the simplest NN, where the input data go to the hidden layer and from there to the output layer. Backpropagation neural network (BPNN) consists of feedforward and weight adjustment of the layers and is the most commonly applied ANN in transportation management. Xu et al. [31] applied BPNN to predict traffic flow, thus to evaluate congestion factor in their study. They proposed occupancy-based congestion factor (CRO) evaluation method with three other evaluated congestion factors based on mileage ratio of congestion (CMRC), road speed (CRS), and vehicle density (CVD). They also evaluated the effect of data-size on real-time rendering of road congestion. Complex road network with higher interconnections showed higher complication in simulation and rendering. The advantage of the proposed model was that it took little processing time for high sampling data rendering. The model can be used as a general congestion prediction model for different road networks. Some used hybrid NN for congestion prediction. Nadeem and Fowdur [11] predicted congestion in spatial space, applying the combination of one of six SML algorithms with NN. Six SML algorithms included moving average (MA), autoregressive integrated moving average (ARIMA), linear regression, second- and third-degree polynomial regression, and k-nearest neighbour (KNN). The model showing the least RMSE value was combined with BPNN to form hybrid NN. The hidden layer had seven neurons, which was determined by trial and error. However, it was a very preliminary level work. It did not show the effect of data increment in the accuracy.

Unlike the previous studies, those focused on traffic flow parameters to conduct traffic congestion prediction research; Ito and Kaneyasu [60] analysed drivers' behaviour in predicting congestion. They showed that vehicle operators act differently on different phases of the journey. They used one layered BPNN to learn the behaviour of female drivers and extract travel phase according to that. The results showed an average efficiency of 82% in distinguishing the travel phase.

ANN is a useful machine learning model which has a flexible structure. The neurons of the layer can be adapted according to the input data. As mentioned above, a general model can be developed and applied for different road types by using the advantage of nonlinearity capturing ability of ANN. However, ANN requires larger datasets than the probabilistic reasoning models, which results in high complexity.

ANN shows great potential in diverse parameter analysis. ANN is the only model that has recently been applied for driver behaviour analysis for traffic congestion. ANN is popular in every section of transport- traffic flow prediction [61, 62], congestion control [63], driver tiredness [64], and vehicle noise [65, 66].

4.2.2. Regression Model. Regression is a statistical supervised ML algorithm. It models the prediction real numbered output value based on the independent input numerical variable. Regression models can be further divided according to the number of input variables. The simplest regression model is linear regression with one input feature. When the feature number increases, the multiple regression model is generated.

Jiwan et al. [27] developed a multiple linear regression analysis (MLRA) model using weather data and traffic congestion data after preprocessing using Hadoop. At first, a single regression model was developed for all the variables using R. After a 3-fold reduction process, only ten variables were determined to form the final MLRA model. Zhang and Qian [22] conducted an interesting approach to predict morning peak hour congestion using household electricity usage patterns. They used LASSO regression to correlate the pattern features using the advantage of linearly related critical feature selection capability.

On the other hand, Jain et al. [33] developed both linear and exponential regression model using IBM SPSS software to find the relevant variables. The authors converted heterogenous vehicles into passenger car unit (PCU) for simplification. Three independent variables were considered to estimation origin-destination- (O-D-) based congestion measures. They used PCC to evaluate the correlation among the parameters. However, simply averaging O-D node parameters may not provide the actual situation of dynamic traffic patterns.

Regression models consist of some hidden coefficients, which are determined in the training phase. The most applied regression model is the autoregressive integrated moving average (ARIMA). ARIMA has three parameters- p, d, and q. "p" is the auto regressive order that refers to how

Table 1: Traffic congestion prediction studies in probabilistic reasoning.

Methodology	Road type	Data source	Input parameters	Target domain	No. of congestion state levels*	Reference
Hierarchical fuzzy rule-based system	Highway corridor	Sensor	Occupancy	Speed	2	Zhang et al. [30]
			Speed	Speed		Lopez-garcia et al. [37]
Evolutionary fuzzy rule learning			Traffic flow	Traffic density		Onieva et al. [28]
Mamdani-type fuzzy logic inference	Highway, trunk road, branch road	—	Speed	Congestion Index	4	Cao and Wang [3]
			Density			
Fuzzy inference	Highway corridor	Camera	Travel time Traffic flow Speed			Wang et al. [58]
Fuzzy comprehensive evaluation	Highway corridor	Probe	Traffic volume Speed	Saturation Density speed	5	Kong et al. [4] Yang et al. [5]
Hidden Markov model	Highway network	Sensor	Emission matrix	Traffic pattern selection	—	Zaki et al. [32]
			Emission matrix Transition matrix	Traffic pattern determination	—	Zaki et al. [25]
	Main road	Probe	Observation probability Transition probability	Mapping GPS data	—	Sun et al. [45]
Gaussian distribution	Highway corridor	Sensor	Traffic volume	Optimal feature selection	—	Yang [29]
Bayesian network	Build-up area	Simulation	Road and bus increment	Congestion probability	—	Yi Liu et al. [59]
	Bridge	Sensor	Intensity Occupation Average speed Average distance			Asencio-Cortés et al. [54]
	Highway network	Sensor	Network direction Day and time weather Incidents Traffic flow Occupancy Speed Level of service Congestion state	Congestion probability		Kim and Wang [34]
Extended Kalman filter	Highway	Camera	Travel time	Data fusion	—	Adetiloye and Awasthi [7]

The table accumulates the data source, scope of the study area, input and resulting parameters, and how many cognitive traffic states were considered in the studies.* 2 = free/congested, 4 = free/light/medium/severe, 5 = very free/free/light/medium/severe

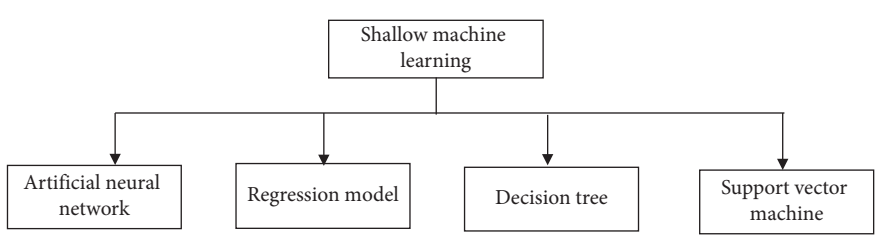


Figure 6: Subdivision of shallow machine learning models.

many lags of the independent variable needs to be considered for prediction. Moving average order "q" presents the lag prediction error numbers. Lastly, "d" is used to make the time-series stationary. Alghamdi et al. [67] took d as 1 as one differencing order could make the model stationary. Next, they applied the autocorrelation function (ACF) and

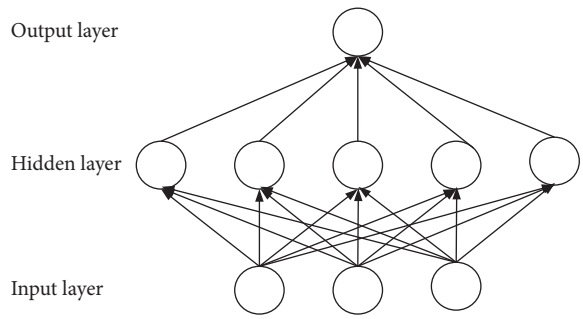


Figure 7: A simple ANN structure.

the partial autocorrelation function (PACF) along with the minimum information criteria matrix to determine the values of p and q. They only took the time dimension into account. However, the results inclined with the true pattern for only one week and needed to be fine-tuned considering prediction errors. Besides, the study did not consider the spatial dimension.

Regression models are useful to be applied for time series problems. Therefore, regression models are suitable for traffic forecasting problems. However, these models are not reliable for nonlinear, rapidly changing the multidimension dataset. The results need to be modified according to prediction errors.

However, as already and further will be discussed in this article, most of the studies used different regression models to validate their proposed model [6, 11, 25, 68, 69].

With the increment of dataset and complexity associated with it, regression models are becoming less popular in traffic congestion prediction. Currently, regression models are frequently used by modifying with other machine learning algorithms, e.g., ANN and kernel functions. Some other sectors' regression models are applied including hybrid ARIMA in traffic speed prediction for specific vehicle type (Wang et al. [70], traffic volume prediction [71], and flow prediction applying modified ARIMA [72].

4.2.3. Decision Tree. A decision tree is a model that predicts an output based on several input variables. There are two types of trees: the classification tree and the regression tree. When these two trees merge, a new tree named classification and regression tree (CART) generates. Decision tree uses the features extracted from the entire dataset. Random forest is a supervised ML classification algorithm that is the average of multiple decision tree results. The features are randomly used while developing decision trees. It uses a vast amount of CART decision trees. The decision trees vote for the predicted class in a random forest model.

Wang et al. [9] proposed a probabilistic method of exploiting information theory tools of entropy and Fano's inequality to predict road traffic pattern and its associated congestion for urban road segments with no prior knowledge on the O-D of the vehicle. They incorporated road congestion level into time series for mapping the vehicle state into the traffic conditions. As interval influenced the predictability, an optimal segment length and velocity was found. However, with less available data, an increased number of segments increased the predictability. Another traffic parameter, travel time, was used to find CI by Liu and Wu [73]. They applied the random forest ML algorithm to forecast traffic congestion states. At first, they extracted 100 sample sets to construct 100 decision trees by using bootstrap. The number of feature attributes was determined as the square root of the total number of features. Chen et al. [16] also applied the CART method for prediction and classification of traffic congestion. The authors applied Moran's I method to analyse the spatiotemporal correlation among different road network traffic flow. The model showed effectiveness compared with SVM and K-means algorithm.

Decision tree is a simple classification problem-solving model that can be applied for multifeature data, e.g., Liu and Wu [73] applied weather condition, road condition, time period, and holiday as the input variables. This model's knowledge can be represented in the form of IF-THEN rules, making it an easily interpretable problem. It is also needed to be kept in mind that the classification results are usually binary and therefore, not suitable where the congestion level is required to be known. Other sectors of transport, where decision tree models applied are traffic prediction [74] and traffic signal optimisation with Fuzzy logic [75].

4.2.4. Support Vector Machine. The support vector machine (SVM) is a statistical machine learning method. The main idea of this model is to map the nonlinear data to a higher dimensional linear space where data can be linearly classified by hyperplane [1]. Therefore, it can be very useful in traffic flow pattern identification for traffic congestion prediction. Tseng et al. [13] determined travel speed in predicting real-time congestion applying SVM. They used Apache Storm to process big data using spouts and bolts. Traffic, weather sensors, and events collected from social media of close proximity were evaluated together by the system. They categorised vehicle speed into classes and referred them as labels. Speed of the previous three intervals was used to train the proposed model. However, the congestion level categororised from 0 to 100 does not carry a specific knowledge of the severity of the level, especially to the road users. Increment in training data raised accuracy and computational time. This may ultimately make it difficult to make real-time congestion prediction.

Traffic flow shows different patterns based on the traffic mixture or time of the day. SVM is applied to identify the appropriate pattern. Currently modified SVM mostly has its application in other sectors as well, e.g., freeway exiting traffic volume prediction [58], traffic flow prediction [76], and sustainable development of transportation and ecology [77].

Most of the studies compared their developed model with SVM [22, 78, 79]. Deep machine learning (DML) algorithms showed better results compared to SVM. Table 2 refers to the studies under this section.

Table 2: Traffic congestion prediction studies in shallow machine learning.

Methodology	Road type	Data source	Input parameters	Target domain	No. of congestion state levels	Reference
Artificial neural network	Road network	Sensor Simulation	Occupancy Density	Congestion factor	3	Xu et al. [31]
	Highway corridor	Sensor	Distance Speed	Speed	2	Nadeem and Fowdur [11]
		Simulation	Speed Throttle opening Steering input angle	Traffic congestion state	2	Ito and Kaneyasu [60]
Regression model	Highway corridor	Sensor	Temperature Humidity Rainfall Traffic speed	Traffic congestion score	—	Jiwan et al. [27]
	Arterial road Subarterial road	Camera	Time Volume Speed	Congestion Index	4	Jain et al. [33]
Decision tree	Ring road	Probe	Average speed	Traffic predictability	—	Wang et al. [9]
	Road network		Speed Trajectory	Moran index	5	Chen et al. [16]
Support vector machine	Highway corridor	Sensor	Speed Density Traffic volume difference Rainfall volume	Travel speed	—	Tseng et al. [13]

4.3. Deep Machine Learning. DML algorithms consist of several hidden layers to process nonlinear problems. The most significant advantage of these algorithms is they can extract features from the input data without any prior knowledge. Unlike SML, feature extraction and model training are done together in these algorithms. DML can convert the vast continuous and complex traffic data with limited collection time horizon into patterns or feature vectors. From last few years, DML has become popular in traffic congestion prediction studies. Traffic congestion studies that used DML algorithms are shown in Figure 8 and discussed in this section.

4.3.1. Convolutional Neural Network. Convolutional neural network (CNN) is a commonly applied DML algorithm in traffic engineering. Due to the excellent performance of CNN in image processing, while applying in traffic prediction, traffic flow data are converted into a 2-D matrix to process. There are five main parts of a CNN structure in transportation: the input layer, convolution layer, pool layer, full connection layer, and output layer. Both the convolution and pooling layer extracts important features. The depth of these two layers differs in different studies. Majority of the studies converted traffic flow data into an image of a 2-D matrix. In the studies performed by Ma et al. [80] and Sun et al. [45], each component of the matrix represented average traffic speed on a specific part of the time. While tuning CNN parameters, they selected a convolutional filter size of (3×3) and max-pooling of size (2×2) of 3 layers according to parameter settings of LeNet and AlexNet and loss of information measurement. Whereas Chen et al. [68] used a five-layered convolution of filter size of (2×2) without the pooling layer. The authors applied a novel method called convolution-based deep neural network modelling periodic traffic data (PCNN). The study folded the time-series to generate the input combining real-time and historical traffic data. To capture the correlation of a new time slot with the immediate past, they duplicated the congestion level of the last slot in the matrix. Zhu et al. [49] also applied five convolution-pooling layers as well as (3×3) and (2×2) sizes, respectively. Along with temporal and spatial data, the authors also incorporated time interval data to produce a 3-D input matrix. Unlike these studies, Zhang et al. [6] preprocessed the raw data by performing a spatiotemporal cross-correlation analysis of traffic flow sequence data using PCC. Then, they applied a model named spatiotemporal feature selection algorithm (STFSA) on the traffic flow sequence data to select the feature subsets as the input matrix. A 2-layered CNN with the convolutional and pooling size as same as the previous studies was used. However, STFSA considers its heuristics, biases, and trade-offs and does not guarantee optimality.

CNN shows good performance, where a large dataset is available. It has excellent feature learning capability with less time-consuming classification ability. Therefore, CNN can be applied where the available dataset can be converted into an image. CNN is applied in traffic speed prediction [81], traffic flow prediction [6], and modified CNN with LSTM is also applied for traffic prediction [82]. However, as mentioned above, no model depth and parameter selection strategies are available.

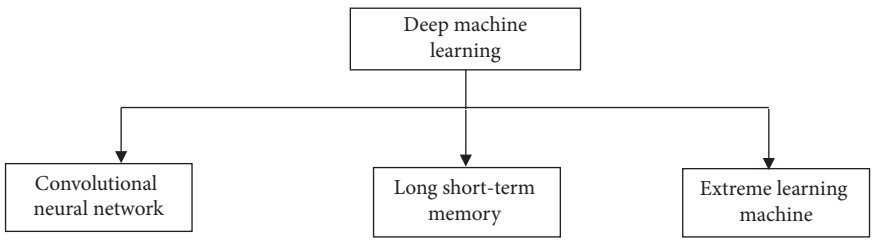


Figure 8: Subdivision of deep machine learning models.

4.3.2. Recurrent Neural Network. Recurrent neural network (RNN) has a wide usage in the sequential traffic data processing by considering the influence of the related neighbour (Figure 9). Long short-term memory (LSTM) is a branch of RNN. In the hidden layer of LSTM, there is a memory block that includes four NN layers, which stores and regulates the information flow. In recent years, with different data collection systems with extended intervals, LSTM has become popular. Due to this advantage, Zhao et al. [12] developed an LSTM model consisted of three hidden layers and ten neurons using long interval data. They set an adequate target and fine-tuned the parameters until the training model stabilized. The authors also applied the congestion index and classification (CI-C) model to classify the congestion by calculating CI from LSTM output data. Most of the studies use the equal interval of CI to divide congestion states. This study did two more intervals of natural breakpoint and geometric interval to find that the latest provided the most information from information entropy. Lee et al. [69] applied 4 layers with 100 neurons LSTM model of 3D matrix input. The input matrix element contained a normalised speed to shorten the training time. While eliminating the dependency, the authors found that a random distribution of target road speed and more than optimally connected roads in the matrix reduced the performance. To eliminate the limitation of temporal dependency, Yuan-Yuan et al. [79] trained their model in the batch learning approach. The instance found from classifying test dataset was used to train the model in an online framework. Some studies introduced new layers to modify the LSTM model for feature extraction. Zhang et al. [83] introduced an attention mechanism layer between LSTM and prediction layer that enabled the feature extraction from a traffic flow data sequence and captured the importance of a traffic state. Di et al. [84] introduced convolution that provides an input to the LSTM model to form the CPM-ConvLSTM model. All the studies applied the one-hot method to convert the input parameters. Adam, stochastic gradient descent (SGD), and leakage integral echo state network (LiESN) are a few optimisation methods applied to fine-tune the outcome.

A few studies combined RNN with other algorithms while dealing with vast parameters of the road network. In this regard, Ma et al. [85] applied the RNN and restricted Boltzmann machine (RNN-RBM) model for networkwide spatiotemporal congestion prediction. Here, they used conditional RBM to construct the proposed deep architecture, which is designed to process the temporal sequence by providing a feedback loop between visible layer and hidden layer. The congestion state was analysed from traffic speed and was represented in binary format in a matrix as input. Also, Sun et al. [45] combined RNN of three hidden layers, with its two other variants: LSTM and gated recurrent unit (GRU). The hidden layers included the memory block characteristic of LSTM, and the cell state and hidden state were incorporated by GRU.

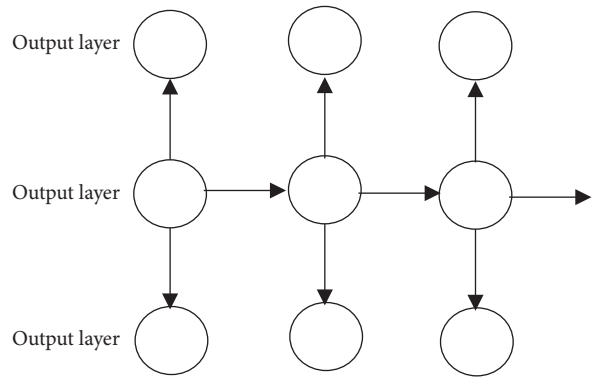


Figure 9: A simple RNN structure.

As the sample size is increasing vastly, RNN is becoming popular as a current way of modelling. RNN has a short-term memory. This characteristic of RNN helps to model nonlinear time series data. The training of RNN is also straight forward, similar to multilayer FNN. However, this training may become difficult due to the conversion in a deep architecture with multiple layers in long-term dependency. In case of long-term dependency problems, LSTM is becoming more suitable to be applied as LSTM can remember information for a long period of time. RNN has its application in other sectors of the transport too, e.g., traffic passenger flow prediction [86], modified LSTM in real time crash prediction [87], and road-network traffic prediction [88].

4.3.3. Extreme Learning Machine. In recent years, a novel learning algorithm called the extreme learning machine (ELM) is proposed for training the single layer feed-forward neural network (SLFN). In ELM, input weights and hidden biases are assigned randomly instead of being exhaustively tuned. Therefore, ELM training is fast. Therefore, taking this advantage into account, Ban et al. [19] applied the ELM model for real-time traffic congestion prediction. They determine CI using the average travel speed. A 4-fold cross-validation was done to avoid noise in raw data. The model

Table 3: Traffic congestion prediction studies in deep machine learning.

Methodology	Road type	Data source	Input parameters	Target domain	No. of congestion state levels	Reference
Convolutional neural networks	Road network	Probe	Average traffic speed	Speed	3	Ma et al. [80]
				Average traffic speed	5	Sun et al. [45]
		Camera	Congestion level	Congestion level	3	Chen et al. [68]
	Highway corridor	Sensor	Traffic flow	Traffic flow	—	Zhang et al. [93]
	Road section	Probe	Weather data Congestion time	Congestion time	5	Zhao et al. [12]
	Arterial road	Online	Congestion level	Congestion level		Yuan-Yuan et al. [79]
Recurrent neural network	Road network	Camera	Spatial similarity feature	Speed	3	Lee et al. [69]
		Sensor	Speed			
		Survey	Peak hour			
	Highway corridor	Sensor	Speed Travel time Volume	Congestion level	4	Zhang et al. [83]
Extreme learning machine	Road network	Probe	Congestion state Current time Road traffic state cluster Last congestion index Road type Number of adjacent roads	Congestion state	2	Ma et al. [85]
				Congestion Index	—	Ban et al. [19]

found optimal hidden nodes to be 200 in terms of computational cost in the study. An extension of this study was done by Shen et al. [78] and Shen et al. [89] by applying a kernel-based semisupervised extreme learning machine (kernel-SSELM) model. This model can deal with the unlabelled data problem of ELM and the heterogenous data influence. The model integrated small-scaled labelled data of transportation personnel and large-scaled unlabelled traffic data to evaluate urban traffic congestion. ELM speeded up the processing time, where kernel function optimized the accuracy and robustness of the whole model. However, real-time labelled data collection was quite costly in terms of human resources and working time, and the number of experts for congestion state evaluation should have been more. Another modification of EML was applied by Yiming et al. [20]. They applied asymmetric extreme learning machine cluster (S-ELM-cluster) model for short-term traffic congestion prediction by determining the CI. The authors divided the study area and implemented submodels processing simultaneously for fast speed.

The ELM model has the advantage in processing large scale data learning at high speed. ELM works better with labelled data. Where both labelled and unlabelled data are available, semisupervised ELM has shown good prediction accuracy, as it was seen from the studies. Other sectors where ELM was applied included air traffic flow prediction [90], traffic flow prediction [91], and traffic volume interval prediction [92].

Other than the models already discussed, Zhang et al. [93] proposed a deep autoencoder-based neural network model with symmetry of four layers for the encoder and the decoder to learn temporal correlations of a transportation network. The first component encoded the vector representation of historical congestion levels and their correlation. They then decoded to build a representation of congestion levels for the future. The second component of DCPS used two dense layers; those converted the output from the decoder to calculate a vector representation of congestion level. However, the process lost information as the congestion level of all the pixels was averaged. This approach needed high iteration and was computationally expensive as all the pixels regardless of roads were considered. Another study applied a generalised version of recurrent neural network named recursive neural network. The difference between these two is, in recurrent NN, weights are shared along the data sequence. Whereas recursive NN is a single neuron model; therefore, weights are shared at every node. Huang et al. [94] applied a recursive NN algorithm named echo state network (ESN). This model consists of an input layer, reservoir network, and output layer. The reservoir layer constructs the rules that connected prediction origin and forecasting horizon. As the study took a large study area with vast link number, they simplified the training rule complexity applying recursive NN. Table 3 summarises some studies.

5. Discussion and Research Gaps

Research in traffic congestion prediction is increasing exponentially. Among the two sources, most of the studies used stationary sensor/camera data. Although sensor data cannot capture the dynamic traffic change, frequent change

in source makes it complicated to evaluate the flow patterns for probe data [95]. Data collection horizon is an important factor in traffic congestion studies. The small horizon of a few days [3] cannot capture the actual situation of the congestion as traffic is dynamic. Other studies that used data for a few months showed the limitation of seasonality [22, 67].

The condition of the surrounding plays an important factor in traffic congestion. A few studies focused on these factors. Two studies considered social media contribution in input parameter [7, 13], and five considered weather condition [12, 13, 27, 34, 73]. Events, e.g., national event, school holiday, and popular sports events, play a big role in traffic congestion. For example, Melbourne, Australia, has two public holidays before and during two most popular sports events of the country. The authorities close a few traffic routes to tackle the traffic and the parade, resulting in traffic congestion. Therefore, more focus must be put in including these factors while forecasting.

Dealing with missing data is a challenge in the data processing. Some excluded the respective data altogether [29], others applied different methods to retrieve the data [59, 85], and some replaced with other data [45]. Missing data imputation can be a useful research scope in transportation engineering.

Machine learning algorithm, especially DML models, is developed with time. This shows a clear impact on the rise of their implementation in traffic congestion forecasting (Figure 10).

Probabilistic reasoning algorithms were mostly applied for a part of the prediction model, e.g., map matching and optimal feature number selection. Fuzzy logic is the most widely used algorithm in this class of algorithms. From other branches, ANN and RNN are the mostly applied models. Most of the studies that applied hybrid or ensembled models belong to probabilistic and shallow learning class. Only two studies applied hybrid deep learning models while predicting networkwide congestion. Tables 4, 5–6 summarize the advantage and weaknesses of the algorithms of different branches.

Among all DML models, RNN is more suitable for time series prediction. In a few studies, RNN performed better than CNN as the gap between the traffic speeds in different classes was very small [12, 69]. However, due to little research in traffic congestion field, a lot of new ML algorithms are yet to be applied.

SML models showed better results than DML while forecasting traffic congestion in the short-term, as SML can process linearity efficiently and linear features have more contribution to traffic flow in short-term. All the short-term forecasting studies discussed in this article applying SML showed promising results. At the same time, DML models showed good accuracy as these models can handle both linear and nonlinear features efficiently. Besides, real-time congestion prediction cannot afford high computation time. Therefore, models taking a short computational time are more effective in this case.

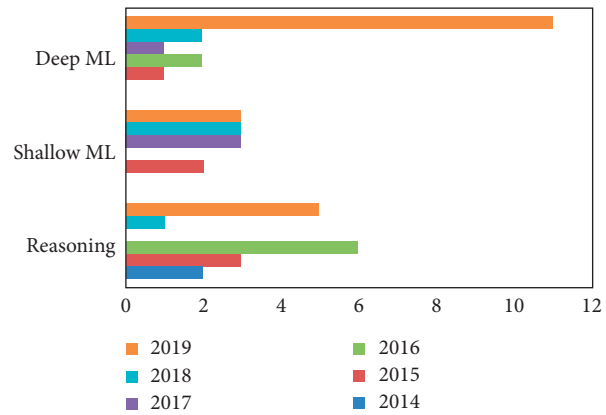


Figure 10: Application of AI models with time.

6. Future Direction

Traffic congestion is a promising area of research. Therefore, there are multiple directions to conduct in future research.

Numerous forecasting models have already been applied in road traffic congestion forecasting. However, with the newly developed forecasting models, there is more scope to make the congestion prediction more precise. Also, in this era of information, the use of increased available traffic data by applying the newly developed forecasting models can improve the prediction accuracy.

The semisupervised model was applied only for the EML model. Other machine learning algorithms should be explored for using both labelled and unlabelled data for higher prediction accuracy. Also, a limited number of studies have focused on real-time congestion forecasting. In future, researches should pay attention to real-time traffic congestion estimation problem.

Another future direction can be focusing on the level of traffic congestion. A few studies have divided the traffic congestion into a few states. However, for better traffic management, knowing the grade of congestion is essential. Therefore, future researches should focus on this. Besides, most studies focused on only one traffic parameter to forecast congestion for congestion prediction. This can be an excellent future direction to give attention to more than one parameter and combining the results during congestion forecasting to make the forecasting more reliable.

Table 4: The strength and weakness of the models of probabilistic reasoning.

Methodology	Advantages	Disadvantages
Fuzzy logic	(i) It converts the binary value into the linguistic description hence portraying the traffic congestion state.	(i) No appropriate membership function shape selection method exists.
	(ii) iIt can portray more than two states.	(ii) Traffic pattern recognition capability is not as durable as ML algorithms.
	(iii) As it does not need an exact crisp input, it can deal with uncertainty.	(iii) Traffic state may not match the actual traffic state as the outcome is not exact.
Hidden Markov model	(i) The model can overcome noisy measurements.	(i) Accuracy decreases with scarce temporal probe trajectory data
	(ii) Can efficiently learn from non-preprocessed data.	(ii) Not suitable in case of missing dataset.
	(iii) Can evaluate multiple hypotheses of the actual mapping simultaneously.	
Gaussian mixture model	(i) Can do traffic parameter distribution over a period as a mixture regardless of the traffic state.	(i) Optimization algorithm used with GMM must be chosen cautiously.
	(ii) Can overcome the limitation of not being able to account for multimodal output by a single Gaussian process.	(ii) Results may show wrong traffic patterns due to local optima limitation and lack of traffic congestion threshold knowledge of the optimisation algorithm.
Bayesian network	(i) It can understand the underlying relationship between random variables.	(i) Computationally expensive.
	(ii) It can model and analyse traffic parameters between adjacent road links.	(ii) The model performs poorly with the increment in data.
	(iii) The model can work with incomplete data.	(iii) The model represents one-directional relation between variables only.

Table 5: The strength and weakness of the models of shallow machine Learning.

Methodology	Advantages	Disadvantages
Artificial neural network	(i) It is an adaptive system that can change structure based on inputs during the learning stage [96].	(i) BPNN requires vast data for training the model due to the parameter complexity resulting from its parameter nonsharing technique [97].
	(ii) It features defined early, FNN shows excellent efficiency in capturing the nonlinear relationship of data.	(ii) The training convergence rate of the model is slow.
Regression model	(i) Models are suitable for time series problems.	(i) Linear models cannot address nonlinearity, making it harder to solve complex prediction problems.
	(ii) Traffic congestion forecasting problems can be easily solved.	(ii) Linear models are sensitive to outliers.
	(iii) ARIMA can increase accuracy by maintaining minimum parameters.	(iii) Computationally expensive.
	(iv) Minimum complexity in the model.	(iv) ARIMA cannot deal multifeature dataset efficiently.
		(v) ARIMA cannot capture the rapidly changing traffic flow [8].
Support vector machine	(i) It is efficient in pattern recognition and classification.	(i) The improperly chosen kernel function may result in an inaccurate outcome.
	(ii) A universal learning algorithm that can diminish the classification error probability by reducing the structural risk [1].	(ii) Unstable traffic flow requires improved prediction accuracy of SVM.
	(iii) It does not need a vast sample size.	(iii) It takes high computational time and memory.

Table 6: The strength and weakness of the models of deep machine learning.

Methodology	Advantages	Disadvantages
Convolutional neural networks	(i) Capable of learning features from local connections and composing them into high-level representation. (ii) Classification is less time-consuming. (iii) Can automatically extract features.	(i) Computationally expensive as a huge kernel is needed for feature extraction. (ii) A vast dataset is required. (iii) Traffic data needs to be converted to an image. (iv) No available strategies are available on CNN model depth and parameter selection.
Recurrent neural network	(i) Shows excellent performance in processing sequential data flow. (ii) Efficient in sequence classification. (iii) Efficient in processing time-series with long intervals and postponements.	(i) Long-term dependency results in bad performance. (ii) No available firm guideline in dependency elimination.
Extreme learning machine	(i) Fast learning speed (ii) Can avoid local minima. (iii) Modified models are available to deal with an unlabelled data problem.	(i) Training time increases with the hidden node rise. (ii) Unlabelled data problem. (iii) May produce less accurate results.

7. Conclusions

Traffic congestion prediction is getting more attention from the last few decades. With the development of infrastructure, every country is facing traffic congestion problem. Therefore, forecasting the congestion can allow authorities to make plans and take necessary actions to avoid it. The development of artificial intelligence and the availability of big data have led researchers to apply different models in this field. This article divided the methodologies in three classed. Although probabilistic models are simple in general, they become complex while different factors that affect traffic congestion, e.g., weather, social media, and event, are considered. Machine learning, especially deep learning, has the benefit in this case. Therefore, deep learning algorithms became more popular with time as they can assess a large dataset. However, a wide range of machine learning algorithms are yet to be applied. Therefore, a vast opportunity of research in the field of traffic congestion prediction still prevails.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank RMIT University and the Australian Government Research Training Program (RTP) for the financial support.

References

[1] Z. Shi, *Advanced Artificial Intelligence*, World Scientific, Singapore, 2011.

[2] M. S. Ali, M. Adnan, S. M. Noman, and S. F. A. Baqueri, "Estimation of traffic congestion cost-A case study of a major arterial in karachi," *Procedia Engineering*, vol. 77, pp. 37–44, 2014.

[3] W. Cao and J. Wang, "Research on traffic flow congestion based on Mamdani fuzzy system," *AIP Conference Proceedings*, vol. 2073, 2019.

[4] X. Kong, Z. Xu, G. Shen, J. Wang, Q. Yang, and B. Zhang, "Urban traffic congestion estimation and prediction based on floating car trajectory data," *Future Generation Computer Systems*, vol. 61, pp. 97–107, 2016.

[5] Q. Yang, J. Wang, X. Song, X. Kong, Z. Xu, and B. Zhang, "Urban traffic congestion prediction using floating car trajectory data," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 18–30, Springer, Zhangjiajie, China, November 2015.

[6] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[7] T. Adetiloye and A. Awasthi, "Multimodal big data fusion for traffic congestion prediction," *Multimodal Analytics for Next-Generation Big Data Technologies and Applications*, Springer, Berlin, Germany, pp. 319–335, 2019.

[8] F. Wen, G. Zhang, L. Sun, X. Wang, and X. Xu, "A hybrid temporal association rules mining method for traffic congestion prediction," *Computers & Industrial Engineering*, vol. 130, pp. 779–787, 2019.

[9] J. Wang, Y. Mao, J. Li, Z. Xiong, and W.-X. Wang, "Predictability of road traffic and Congestion in urban areas," *PLoS One*, vol. 10, no. 4, Article ID e0121825, 2015.

[10] Z. He, G. Qi, L. Lu, and Y. Chen, "Network-wide identification of turn-level intersection congestion using only low-frequency probe vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 108, pp. 320–339, 2019.

[11] K. M. Nadeem and T. P. Fowdur, "Performance analysis of a real-time adaptive prediction algorithm for traffic congestion," *Journal of Information and Communication Technology*, vol. 17, no. 3, pp. 493–511, 2018.

[12] H. Zhao, X. Jizhe, L. Fan, L. Zhen, and L. Qingquan, "A peak traffic Congestion prediction method based on bus driving time," *Entropy*, vol. 21, no. 7, p. 709, 2019.

[13] F.-H. Tseng, J.-H. Hsueh, C.-W. Tseng, Y.-T. Yang, H.-C. Chao, and L.-D. Chou, "Congestion prediction with big data for real-time highway traffic," *IEEE Access*, vol. 6, pp. 57311–57323, 2018.

[14] D. Li, J. Deogun, W. Spaulding, and B. Shuart, "Towards missing data imputation: a study of fuzzy K-means Clustering method," in *Rough Sets and Current Trends in Computing*, S. Tsumoto, R. Słowiński, J. Komorowski, and J. W. Grzymała-Busse, Eds., pp. 573–579, Springer, Berlin, Germany, 2004.

[15] Y. Yang, Z. Cui, J. Wu, G. Zhang, and X. Xian, "Fuzzy c-means clustering and opposition-based reinforcement learning for traffic congestion identification," *Journal of Information & Computational Science*, vol. 9, no. 9, pp. 2441–2450, 2012.

[16] Z. Chen, Y. Jiang, D. Sun, and X. Liu, "Discrimination and prediction of traffic congestion states of urban road network based on spatio-temporal correlation," *IEEE Access*, vol. 8, pp. 3330–3342, 2020.

[17] Y. Guo, L. Yang, S. Hao, and J. Gao, "Dynamic identification of urban traffic congestion warning communities in heterogeneous networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 522, pp. 98–111, 2019.

[18] A. Elfar, A. Talebpour, and H. S. Mahmassani, "Machine learning approach to short-term traffic congestion prediction in a connected environment," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 45, pp. 185–195, 2018.

[19] X. Ban, C. Guo, and G. Li, "Application of extreme learning machine on large scale traffic congestion prediction," *Proceedings of ELM-2015*, vol. 1, pp. 293–305, 2016.

[20] X. Yiming, B. Xiaojuan, L. Xu, and S. Qing, "Large-scale traffic Congestion prediction based on the symmetric extreme learning machine Cluster fast learning method," *Symmetry*, vol. 11, no. 6, p. 730, 2019.

[21] Y. Zheng, Y. Li, C.-M. Own, Z. Meng, and M. Gao, "Real-time predication and navigation on traffic congestion model with equilibrium Markov chain," *International Journal of Distributed Sensor Networks*, vol. 14, no. 4, Article ID 155014771876978, 2018.

[22] P. Zhang and Z. Qian, "User-centric interdependent urban systems: using time-of-day electricity usage data to predict morning roadway congestion," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 392–411, 2018.

[23] P. Mishra, R. Hadfi, and T. Ito, "Adaptive model for traffic congestion prediction," in *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, vol. 9799, pp. 782–793, Morioka, Japan, 2016.

[24] F. Li, J. Gong, Y. Liang, and J. Zhou, "Real-time congestion prediction for urban arterials using adaptive data-driven methods," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 17573–17592, 2016.

[25] J. F. Zaki, A. Ali-Eldin, S. E. Hussein, S. F. Saraya, and F. F. Areed, "Traffic congestion prediction based on Hidden Markov Models and contrast measure," *Ain Shams Engineering Journal*, vol. 11, no. 3, p. 535, 2020.

[26] P. Jiang, L. Liu, L. Cui, H. Li, and Y. Shi, "Congestion prediction of urban traffic employing SRBDP," in *Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, pp. 1099–1106, IEEE, Guangzhou, China, 2017.

[27] L. Jiwan, H. Bonghee, L. Kyungmin, and J. Yang-Ja, "A prediction model of traffic congestion using weather data," in *Proceedings of the 2015 IEEE International Conference on Data Science and Data Intensive Systems*, pp. 81–88, Sydney, NSW, Australia, December 2015.

[28] E. Onieva, P. Lopez-Garcia, A. D. Masegosa, E. Osaba, and A. Perallos, "A comparative study on the performance of evolutionary fuzzy and Crisp rule based Classification methods in Congestion prediction," *Transportation Research Procedia*, vol. 14, pp. 4458–4467, 2016.

[29] S. Yang, "On feature selection for traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 160–169, 2013.

[30] X. Zhang, E. Onieva, A. Perallos, E. Osaba, and V. C. S. Lee, "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 127–142, 2014.

[31] Y. Xu, L. Shixin, G. Keyan, Q. Tingting, and C. Xiaoya, "Application of data science technologies in intelligent prediction of traffic Congestion," *Journal of Advanced Transportation*, 2019.

[32] J. Zaki, A. Ali-Eldin, S. Hussein, S. Saraya, and F. Areed, "Time aware hybrid hidden Markov models for traffic Congestion prediction," *International Journal on Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 1–17, 2019.

[33] S. Jain, S. S. Jain, and G. Jain, "Traffic congestion modelling based on origin and destination," *Procedia Engineering*, vol. 187, pp. 442–450, 2017.

[34] J. Kim and G. Wang, "Diagnosis and prediction of traffic Congestion on urban road networks using bayesian networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2595, no. 1, pp. 108–118, 2016.

[35] W.-X. Wang, R.-J. Guo, and J. Yu, "Research on road traffic congestion index based on comprehensive parameters: taking Dalian city as an example," *Advances in Mechanical Engineering*, vol. 10, no. 6, Article ID 168781401878148, 2018.

[36] E. Onieva, V. Milanes, J. Villagra, J. Perez, and J. Godoy, "Genetic optimization of a vehicle fuzzy decision system for intersections," *Expert Systems with Applications*, vol. 39, no. 18, pp. 13148–13157, 2012.

[37] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos, "A hybrid method for short-term traffic Congestion forecasting using genetic algorithms and Cross entropy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 557–569, 2016.

[38] A. Daissaoui, A. Boulmakoul, and Z. Habbas, "First specifications of urban traffic-congestion forecasting models," in *Proceedings of the 27th International Conference on Microelectronics (ICM 2015).*, pp. 249–252, Casablanca, Morocco, December 2015.

[39] 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.

[40] M. B. Trabia, M. S. Kaseko, and M. Ande, "A two-stage fuzzy logic controller for traffic signals," *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 6, pp. 353–367, 1999.

[41] Y. Zhang and Z. Ye, "Short-term traffic flow forecasting using fuzzy logic system methods," *Journal of Intelligent Transportation Systems*, vol. 12, no. 3, pp. 102–112, 2008.

[42] H. Wang, L. Zheng, and X. Meng, *Traffic Accidents Prediction Model Based on Fuzzy Logic*, Springer, Berlin, Germany, 2011.

[43] Y. Zhang and H. Ge, "Freeway travel time prediction using takagi-sugeno-kang fuzzy neural network," *Computer-aided Civil and Infrastructure Engineering*, vol. 28, no. 8, pp. 594–603, 2013.

[44] J. Zhao, "Research on prediction of traffic congestion state," in *Proceedings of the MATEC Web of Conferences*, Les Ulis, France, 2015.

[45] S. Sun, J. Chen, and J. Sun, "Traffic congestion prediction based on GPS trajectory data," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, Article ID 155014771984744, 2019.

[46] Y. Qi and S. Ishak, "A Hidden Markov Model for short term prediction of traffic conditions on freeways," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 95–111, 2014.

[47] B. Jiang and Y. Fei, "Traffic and vehicle speed prediction with neural network and hidden markov model in vehicular networks," in *Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1082–1087, IEEE, Seoul, 2015.

[48] G. Zhu, K. Song, P. Zhang, and L. Wang, "A traffic flow state transition model for urban road network based on Hidden Markov Model," *Neurocomputing*, vol. 214, pp. 567–574, 2016a.

[49] L. Zhu, R. Krishnan, F. Guo, J. Polak, and A. Sivakumar, "Early identification of recurrent congestion in heterogeneous urban traffic," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 4392–4397, IEEE, Auckland, New Zealand, October 2019.

[50] Y. Xie, K. Zhao, Y. Sun, and D. Chen, "Gaussian processes for short-term traffic volume forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2165, no. 1, pp. 69–78, 2010.

[51] S. Jin, X. Qu, and D. Wang, "Assessment of expressway traffic safety using Gaussian mixture model based on time to collision," *International Journal of Computational Intelligence Systems*, vol. 4, no. 6, pp. 1122–1130, 2011.

[52] J. Jun, "Understanding the variability of speed distributions under mixed traffic conditions caused by holiday traffic," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 599–610, 2010.

[53] S. Shiliang, Z. Changshui, and Y. Guoqiang, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[54] G. Asencio-Cortés, E. Florido, A. Troncoso, and F. Martínez-Álvarez, "A novel methodology to predict urban traffic congestion with ensemble learning," *Soft Computing*, vol. 20, no. 11, pp. 4205–4216, 2016.

[55] Z. Zhu, B. Peng, C. Xiong, and L. Zhang, "Short-term traffic flow prediction with linear conditional Gaussian Bayesian network," *Journal of Advanced Transportation*, vol. 50, no. 6, pp. 1111–1123, 2016.

[56] S. Wang, W. Huang, and H. K. Lo, "Traffic parameters estimation for signalized intersections based on combined shockwave analysis and Bayesian Network," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 22–37, 2019.

[57] S. Wang, W. Huang, and H. K. Lo, "Combining shockwave analysis and Bayesian Network for traffic parameter estimation at signalized intersections considering queue spillback," *Transportation Research. Part C, Emerging Technologies*, vol. 120, Article ID 102807, 2020.

[58] X. Wang, K. An, L. Tang, and X. Chen, "Short term prediction of freeway exiting volume based on SVM and KNN," *International Journal of Transportation Science and Technology*, vol. 4, no. 3, pp. 337–352, 2015.

[59] Y. Liu, X. Feng, Q. Wang, H. Zhang, and X. Wang, "Prediction of urban road Congestion using a bayesian network approach," *Procedia—Social and Behavioral Sciences*, vol. 138, no. C, pp. 671–678, 2014.

[60] T. Ito and R. Kaneyasu, "Predicting traffic congestion using driver behavior," in *Proceedigs of the International Conference on Knowledge Based and Intelligent Information and Engineering Systems*, Marseille, France, 2017.

[61] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 755–764, 2013.

[62] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction in heterogeneous condition using artificial neural network," *Transport*, vol. 30, no. 4, pp. 397–405, 2013.

[63] R. More, A. Mugal, S. Rajgure, R. B. Adhao, and V. K. Pachghare, "Road traffic prediction and congestion control using artificial neural networks," in *Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pp. 52–57, IEEE, Pune, India, December 2016.

[64] C. Jacobé de Naurois, C. Bourdin, A. Stratulat, E. Diaz, and J.-L. Vercher, "Detection and prediction of driver drowsiness using artificial neural network models," *Accident Analysis & Prevention*, vol. 126, pp. 95–104, 2019.

[65] P. Kumar, S. P. Nigam, and N. Kumar, "Vehicular traffic noise modeling using artificial neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 111–122, 2014.

[66] V. Nourani, H. Gökçekuş, I. K. Umar, and H. Najafi, "An emotional artificial neural network for prediction of vehicular traffic noise," *Science of The Total Environment*, vol. 707, p. 136134, 2020.

[67] 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)*, pp. 1227–1232, IEEE, Tangier, Morocco, October 2019.

[68] M. Chen, X. Yu, and Y. Liu, "PCNN: deep convolutional networks for short-term traffic congestion prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 11, pp. 3550–3559, 2018.

[69] C. Lee, Y. Kim, S. Jin et al., "A visual analytics system for exploring, monitoring, and forecasting road traffic Congestion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3133–3146, 2020.

[70] H. Wang, L. Liu, S. Dong, Z. Qian, and H. Wei, "A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD-ARIMA framework," *Transportmetrica B: Transport Dynamics*, vol. 4, no. 3, pp. 159–186, 2016.

[71] X. F. Wang, X. Y. Zhang, Q. Y. Ding, and Z. Q. Sun, "Forecasting traffic volume with space-time ARIMA model," *Advanced Materials Research*, vol. 156-157, pp. 979–983, 2010.

[72] L. Kui-Lin, Z. Chun-Jie, and X. Jian-Min, "Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN," in *Proceedings of the 2017 Chinese Automation Congress (CAC)*, pp. 2804–2807, Jinan, China, October 2017.

[73] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest,"vol. 2, pp. 361–364, in *Proceedings of the 2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, pp. 361–364, IEEE, Hangzhou, China, December 2017.

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

[75] M. Balta and İ. Özçelik, "A 3-stage fuzzy-decision tree model for traffic signal optimization in urban city via a SDN based VANET architecture," *Future Generation Computer Systems*, vol. 104, pp. 142–158, 2020.

[76] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2019.

[77] S. Lu and Y. Liu, "Evaluation system for the sustainable development of urban transportation and ecological environment based on SVM," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 2, pp. 831–838, 2018.

[78] Q. Shen, X. Ban, and C. Guo, "Urban traffic Congestion evaluation based on kernel the semi-supervised extreme learning machine," *Symmetry*, vol. 9, no. 5, p. 70, 2017.

[79] C. Yuan-Yuan, Y. Lv, Z. Li, and F.-Y. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 132–137, Rio de Janeiro, Brazil, December 2016.

[80] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[81] R. Ke, W. Li, Z. Cui, and Y. Wang, "Two-stream multi-channel Convolutional neural network for multi-lane traffic speed prediction Considering traffic volume impact," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 4, pp. 459–470, 2020.

[82] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.

[83] T. Zhang, Y. Liu, Z. Cui, J. Leng, W. Xie, and L. Zhang, "Short-term traffic Congestion forecasting using attention-based long short-term memory recurrent neural network," in *Proceedings of the International Conference on Computational Science*, Faro, Portugal, June 2019.

[84] X. Di, Y. Xiao, C. Zhu, Y. Deng, Q. Zhao, and W. Rao, "Traffic congestion prediction by spatiotemporal propagation patterns," in *Proceedings of the 2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pp. 298–303, Hong Kong, China, June 2019.

[85] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PloS One*, vol. 10, no. 3, Article ID e0119044, 2015.

[86] Z. Zhene, P. Hao, L. Lin et al., "Deep convolutional mesh RNN for urban traffic passenger flows prediction," in *Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1305–1310, IEEE, Guangzhou, China, October 2018.

[87] P. Li, M. Abdel-Aty, and J. Yuan, "Real-time crash risk prediction on arterials based on LSTM-CNN," *Accident Analysis & Prevention*, vol. 135, Article ID 105371, 2020.

[88] W. Xiangxue, X. Lunhui, C. Kaixun, X. Lunhui, C. Kaixun, and C. Kaixun, "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.

[89] Q. Shen, X. Ban, C. Guo, and C. Wang, "Kernel based semi-supervised extreme learning machine and the application in traffic congestion evaluation," *Proceedings of ELM-2015*, vol. 1, pp. 227–236, 2016.

[90] Z. Zhang, A. Zhang, C. Sun et al., "Research on air traffic flow forecast based on ELM non-iterative algorithm," *Mobile Networks and Applications*, pp. 1–15, 2020.

[91] Y.-m. Xing, X.-j. Ban, and R. Liu, "A short-term traffic flow prediction method based on kernel extreme learning machine," in *Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 533–536, IEEE, Shanghai, China, January 2018.

[92] L. Lin, J. C. Handley, and A. W. Sadek, "Interval prediction of short-term traffic volume based on extreme learning machine and particle swarm optimization," in *Proceedings of the 96th Transportation Research Board Annual*, Washington DC, USA, 2017.

[93] S. Zhang, Y. Yao, J. Hu, Y. Zhao, S. Li, and J. Hu, "Deep autoencoder neural networks for short-term traffic Congestion prediction of transportation networks," *Sensors*, vol. 19, no. 10, p. 2229, 2019.

[94] D. Huang, Z. Deng, S. Wan, B. Mi, and Y. Liu, "Identification and prediction of urban traffic congestion via cyber-physical link optimization," *IEEE Access*, vol. 6, pp. 63268–63278, 2018.

[95] V. Ahsani, M. Amin-Naseri, S. Knickerbocker, and A. Sharma, "Quantitative analysis of probe data characteristics: coverage, speed bias and congestion detection precision," *Journal of Intelligent Transportation Systems*, vol. 23, no. 2, pp. 103–119, 2019.

[96] S. J. Kwon, *Artificial Neural Networks*, Nova Science Publishers, New York, NY, USA, 2011.

[97] Y. Liu, Z. Liu, and R. Jia, "DeepPF: a deep learning based architecture for metro passenger flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 18–34, 2019.

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 8898507, 31 pages
https://doi.org/10.1155/2021/8898507

Research Article

A Decision-Making Method for Ship Collision Avoidance Based on Improved Cultural Particle Swarm

Yisong Zheng, Xiuguo Zhang, Zijing Shang, Siyu Guo, and Yiquan Du

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Xiuguo Zhang; zhangxg@dlmu.edu.cn

Received 21 September 2020; Revised 30 November 2020; Accepted 15 December 2020; Published 15 January 2021

Academic Editor: Petr Dolezel

In the process of ship collision avoidance decision making, steering collision avoidance is the most frequently adopted collision avoidance method. In order to obtain an effective and reasonable steering angle, this paper proposes a decision-making method for ship collision avoidance based on improved cultural particle swarm. Firstly, the ship steering angle direction is to be determined. In this stage, the Kalman filter is used to predict the ship's trajectory. According to the prediction parameters, the collision risk index of the ship is calculated and the situation with the most dangerous ship is judged. Then, the steering angle direction of the ship is determined by considering the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs). Secondly, the ship steering angle is to be calculated. In this stage, the cultural particle swarm optimization algorithm is improved by introducing the index of population premature convergence degree to adaptively adjust the inertia weight of the cultural particle swarm so as to avoid the algorithm falling into premature convergence state. The improved cultural particle swarm optimization algorithm is used to find the optimal steering angle within the range of the steering angle direction. Compared with other evolutionary algorithms, the improved cultural particle swarm optimization algorithm has better global convergence. The convergence speed and stability are also significantly improved. Thirdly, the ship steering angle direction decision method in the first stage and the ship steering angle decision method in the second stage are integrated into the electronic chart platform to verify the effectiveness of the decision-making method of ship collision avoidance presented in this paper. Results show that the proposed approach can automatically realize collision avoidance from all other ships and it has an important practical application value.

1. Introduction

The increasing density of ship traffic flow leads to the frequent occurrence of ship collision accidents. More than 90% of ship collision accidents are related to human factors, of which at least 60% are directly caused by human errors [1]. At present, a large number of advanced auxiliary systems such as Global Positioning System (GPS), Automatic Radar Mapping Assistant (ARPA), Automatic Identification System (AIS), and Electronic Chart Display and Information System (ECDIS) have brought great help to the acquisition of ships during navigation [2]. However, the progress of information acquisition has not reduced the collision probability of ships to the ideal level [3]. On the other hand, due to the increasingly complex collision avoidance environment faced by ships, it is difficult to describe all possible conditions with rules, which requires ships to have the ability to autonomously analyze the surrounding environment so as to make accurate collision avoidance decisions.

Autonomous ship navigation can be divided into two main research areas: collision avoidance and track keeping [4]. Among them, collision avoidance is the core of autonomous ship navigation. The realization of ship collision avoidance first needs to divide the encounter situation of the target ship on the basis of the known environment, quantify the collision risk of the ship, determine the key avoiding ships, then establish the mathematical model of steering, and finally calculate the avoidance decision of own ship in the situation [5]. However, apart from the general rules outlined in the International Regulations for Preventing Collisions at Sea (COLREGs) and the traditional operating specifications, there are no special rules to guide the ship's navigation

system for avoidance manipulation in situations involving multiple ships, and there is no guidance. When multiple ships meet, the collision avoidance decision to avoid collision depends on many factors, such as ship speed, heading, relative position, and maneuverability of the ship [6]. Changing the course of a ship can achieve better collision avoidance than changing the speed. At the same time, it is easier for the surrounding ships to observe the change of the ship's course on vision and radar [7]. Therefore, finding the optimal steering angle of a ship is currently the most frequently adopted collision avoidance decision-making method.

Finding the optimal steering angle is an objective optimization problem. At present, objective optimization problems are divided into linear optimization and nonlinear optimization. Linear optimization problems include the simplex method, interior point method, and polynomial method. The Newton method and conjugate gradient method are commonly used in nonlinear optimization problems. In essence, these optimization problems can be summarized into two steps: the first step is to build a mathematical model based on the actual problem according to its description; The second step is to use a certain method to find the optimal solution within its limited range [8]. The optimal solution of a single-objective optimization problem is generally a solution that can be clearly defined, and the optimal solution of a multiobjective optimization problem is generally an optimal solution set containing multiple optimal solutions [9]. In the objective optimization problem, there are currently two optimization algorithms: deterministic and random algorithms. Deterministic algorithms are mainly based on gradient search methods [10]. The shortcomings of this method are as follows: on the one hand, it has high requirements for the initial value points and gradient information, and on the other hand, it is difficult to optimize the nonderivable and feasible region disconnection. Random algorithms mainly include simulated annealing algorithm (SA) [11], tabu search algorithm (TS) [12], and evolutionary algorithms [13]. The evolutionary algorithm is more robust than SA and TS [14] and also has strong search efficiency; compared with deterministic algorithms, evolutionary algorithms have fewer constraints and are easier to implement when solving constrained optimization problems. With the successful application of evolutionary algorithms in many fields, people's research efforts have also been strengthened, which has also promoted the rapid development of evolutionary algorithms.

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart in 1995 based on the predatory behavior of birds [15], which belongs to an evolutionary algorithm. This algorithm has the following advantages: simple calculation and good robustness. However, there are still some weaknesses in solving the problem of objective optimization, such as easy to fall into local optimum, poor global convergence, and so on. In order to overcome the defects and improve the local optimization ability of PSO algorithm, scholars at home and abroad have proposed many improvement methods for the inertia weight of particles. For example, in [16], a nonlinear decreasing algorithm of inertia weight is proposed. Based on the basic idea of linearly decreasing inertial weight values, three nonlinear decreasing strategies are proposed. In [17], the problem of adaptive change of inertial weight is studied. In [18], the inertia weight change according to the exponential law is proposed. In [19], stochastic inertia weighting strategies are proposed. Compared with the standard PSO algorithm, these methods can improve the convergence speed and optimization accuracy.

However, in these previous improvement studies, the linearly decreasing inertial weight method is usually used, which cannot adapt to complex nonlinear optimization problems, such as finding the optimal steering angle of a ship. Besides, only the optimization and improvement of inertial weights are improved. The algorithm still has the possibility of falling into a local optimum.

In view of the above problems, this paper proposes an improved cultural particle swarm optimization algorithm, which can dynamically and nonlinearly change the inertia weights, and solves the problem of falling into a local optimum. The cultural particle swarm optimization algorithm (CPSO) [20, 21] integrates the PSO algorithm into the framework of cultural algorithms and composes the main group space (lower space) and the knowledge space (upper space) based on PSO. Both levels have their groups and evolve independently and in parallel. The lower space regularly provides excellent individuals to the upper space. After the upper space has evolved, it regularly guides the evolution of the lower space; therefore, a "double evolution and double promotion" mechanism is formed so as to increase the population diversity of PSO and avoid "prematurity" to improve the calculation accuracy and computational efficiency. This article introduces the index of the premature convergence degree of the particle swarm, which is calculated to determine the spatial state of the population, and the inertia weight is dynamically and nonlinearly changed, to improve the optimization efficiency of the cultural particle swarm algorithm. Simulation results prove that in terms of the steering angle of a multitarget ship, the algorithm can effectively avoid the defect of being easily trapped into a local optimum, and the convergence speed and optimization accuracy are far better than the single PSO and CPSO algorithms.

The rest of the paper is organized as follows. The related works are introduced in Section 2. Section 3 gives a ship collision avoidance decision framework based on improved cultural particle swarms. Section 4 presents the steering angle direction decision method based on ship collision risk. Section 5 presents the optimal steering angle decision algorithm based on improved cultural particle swarms. Section 6 integrates the method of this paper with the electronic chart platform and performs experimental analysis. Finally, the summary and conclusions are given in Section 7.

2. Related Research

At present, in the practice of ship collision avoidance, it is still inseparable from the subjective decision of the driver, that is, to manually complete the collision avoidance task

based on experience. In the early stage of the shipping industry, the density of traffic flow is small, and the ship speed is low. The method of subjective judgment and manual operation for collision avoidance decision is still competent. However, as the density of ships traffic increases, it becomes more and more difficult to manually complete the collision avoidance decision task of ships. So far, there has been no effective method to satisfy mariners in the area of ship automation collision avoidance decision, and it has become an urgent problem that needs to be solved. The problem of collision avoidance decision is essential to find the optimal steering angle of the ship without collision. At present, many solutions to this problem have been proposed. The mainstream methods include traditional methods such as ship optimal control theory [22], speed obstacle method [23], distributed tabu search algorithm [24], including evolutionary algorithms [25], and other related methods.

In terms of traditional methods, Johansen et al. [22] proposed a ship collision avoidance model based on the optimal control theory. By changing the offset to the autopilot's heading angle, a limited set of alternative control behaviors can be obtained, and then whether these alternative control behaviors meet the COLREGs or not is determined and the relevant collision risk is evaluated to find the best control behavior. But this method mainly considers the situation of one-to-one ships and does not include the treatment of priority avoidance of the most dangerous target ship in case of multiple target ships (two or more). Kuwata et al. [23] proposed a ship autonomous motion model based on the speed obstacle method. This model combines COLREGs to generate a cone-shaped obstacle in the ship's speed space. It dynamically jumps out of the obstacle range by changing the course of the ship to ensure that the ship's speed vector is not within the range of the cone obstacle. The advantage of this method is that it can check ship collisions along arbitrary trajectories. However, since collision checking needs to be performed on many time slices, it is computationally expensive and time-consuming. Kim et al. [24] used distributed taboo search algorithm (DTSA) to find the best course for the ship. When multiple ships meet, it is assumed that the highest priority of selecting the next course will be given to the ship, which can minimize the risk of collision by changing the course. This method calculates the collision risk from the information of the current heading received by neighboring ships. This process is repeated until the collision risk disappears. Although these works consider a distributed coordination structure involving multiple ships, they ignore the AIS (automatic identification system). The problem that the data cannot be updated on time causes the ship to be unable to change the course according to the prescribed time.

In recent years, the rapid development of intelligent technologies and methods has overcome the difficulties of abstract factors and quantification difficulties encountered in the early use of deterministic methods to solve collision avoidance decision problems [26]. For example, the impact of these parameters on ship collisions estimated only based on existing experiments and expert experience will make the collision avoidance decision unreliable. The fuzzy distribution usually needs to find a function similar to the ship collision within a given fuzzy distribution function range. The actual parameters are determined through prior knowledge or experimental data to obtain the specific membership function and realize the quantification of the collision risk of ships. Therefore, the application of evolutionary algorithms in ship collision avoidance has gradually become a new research direction. For example, Lazarowska et al. [25] applied the ant colony (ACO) algorithm to the unmanned surface vehicle (USV) control system and realized the decision of USV optimal steering angle included in the high seas and restricted waters. The system converts the problem of finding the optimal rudder angle into an optimization problem, takes collision risk and voyage loss as the objective function, and uses the ACO algorithm to solve the optimal solution. However, the convergence speed of the ACO algorithm is slow and it is prone to appear stagnation, and the ship information acquisition error is not considered. Tam et al. [27] thought of all ships as moving points and used the genetic algorithm (GA) to analyze the relevant information of the ships that meet in close range to find the optimal steering angle of the ship. Although the GA has a good global search capability, the operation of the algorithm requires more training time.

In summary, the current mainstream ship collision avoidance methods still have various problems, which are prominently manifested in the following aspects:

(1) Most of the methods of steering and collision avoidance do not consider the problems of delayed data update, equipment operation errors, and other problems leading to untimely data updates and inaccurate ship trajectory, which affects the decision of steering angle.

(2) Most studies focus on the collision avoidance behavior of one-to-one ships and do not analyze the situation of the ship's encounter with the most dangerous target ships, in the case of encounters with multitarget ships (two and more), resulting in the ship's steering angle direction decisions not satisfying COLREGs.

(3) Although the evolutionary algorithm has a good performance in finding the optimal steering angle of the ship without collision, the related evolutionary algorithm is liable to fall into a local optimum or the training time is too long.

Aiming at the above problems, this paper proposes a collision avoidance decision method based on improved cultural particle swarm optimization, which combines Kalman filtering algorithm, fuzzy distribution algorithm, CPSO algorithm, and COLREGs. The main contributions of the method in this paper are as follows:

(1) This method firstly uses the advantages of the Kalman filter algorithm in target tracking and prediction and smoothing and prediction of ship motion trajectory through AIS ship observation node data; the error caused by the AIS data transmission is reduced, the accuracy of the data is improved, and

the accuracy of the collision avoidance decision is further improved. The problem that the AIS data are not updated in time is solved, the accuracy of the data is improved, and the accuracy of collision avoidance decision is further improved.

(2) Using the advantage of fuzzy distribution algorithm in dealing with uncertain mathematical models, this paper establishes the corresponding membership function for several factors that mainly affect the ship's risk degree, solves the ship's collision risk degree, and finds out the most dangerous target ship. Combined with COLREGs, the problem of steering angle direction is solved by analyzing the encounter situation between the ship and the most dangerous target ship when meeting with two or more target ships.

(3) In the calculation of the optimal steering angle, the index to evaluate the premature convergence degree of particle swarm is introduced to dynamically change the inertia weight of particles, and the advantage of cultural particle swarm algorithm in finding the global optimal solution is higher than that of GA, PSO, and other common evolutionary algorithms in calculation accuracy and efficiency, which improves the convergence speed and optimization accuracy of the algorithm and solves the problem that the evolutionary algorithm is easy to fall into local optimum and training time is too long.

3. Ship Collision Avoidance Decision Framework Based on Improved Cultural Particle Swarm

The ship collision avoidance decision framework based on the improved cultural particle swarm is shown in Figure 1. The framework is mainly composed of three stages of functions. In the first stage, the collision risk between the ship and the surrounding target ships is analyzed and the optimal steering angle direction is determined according to COLREGs. In the second stage, the ship's steering angle is calculated based on the steering angle direction determined in the first stage, and the obtained steering angle is input to the ship control module to realize the ship collision avoidance operation. In the third stage, the method of the ship steering angle direction decision in the first stage and the method of the ship steering angle decision in the second stage are integrated into the electronic chart platform, and the validity of the collision avoidance decision method studied in this paper is verified.

3.1. Ship Steering Angle Direction Decision. In this stage, the direction of ship steering angle is determined. Since the ship is on the high seas and the waters connected with it that can be navigable, in addition to the implementation of local rules in ports and rivers, COLREGs will be followed first, that is, before calculating the optimal steering angle of the ship, the direction of ship steering angle needs to be determined according to COLREGs. This stage includes 4 steps: ship trajectory prediction, collision risk calculation, judgment of the situation with the most dangerous ship, and judgment of the steering angle direction.

First, the Kalman filter algorithm is used to smooth and predict the ship's motion trajectory to solve the problem of untimely updating of AIS data. Then, the fuzzy distribution algorithm is used to solve the collision risk of the ship and find the most dangerous target ship. Finally, according to the influence parameters of own ship and the most dangerous target ship, it is divided into 16 kinds of ship encounter situations according to literature [28], and the meeting situation of own ship and the most dangerous target ship is judged. By consulting the captain and crew having practical navigation experience and according to the collision avoidance rules and the encounter situation between the own ship and target ship, the rudder angle direction of the ship is determined.

3.2. Optimal Steering Angle Decision. At this stage, the optimal steering angle of the ship is calculated based on the improved cultural particle swarm algorithm.

First, an index that evaluates the degree of premature convergence of the particle swarm is introduced to determine the state of the particles in the population space to determine the degree of convergence of the population, and then the inertial weight is adaptively changed according to the result of the judgment to ensure the diversity of the particles in the population space. Then, the optimal steering angle is calculated based on the improved cultural particle swarm algorithm. The fitness function is constructed with the help of the collision risk of the ship, and the main group space (lower space) and knowledge space (upper space) of the algorithm are established. The interaction between them improves the evolution efficiency to find the optimal steering angle of the ship more effectively. Finally, the obtained optimal steering angle is input to the ship control module to achieve collision avoidance of the ship.

3.3. Electronic Chart Platform Integration and Display. This paper uses electronic charts as a display and verification platform and integrates the steering angle decision method based on ship collision risk and the optimal steering angle decision method based on improved cultural particle swarm algorithm into the electronic chart platform. The platform can dynamically display the status and trajectory of all ships in real time, which can easily show the experimental results of collision avoidance between own ship and target ship and verify the feasibility of the method.

4. Direction Decision Method of Steering Angle Based on Ship Collision Risk

4.1. Optimal Steering Angle Decision. The real-time acquisition of accurate ship AIS information is the database of ship collision avoidance decision making. Only the real-time acquisition of ship AIS data can effectively monitor the navigation status of the water area and timely find the ship

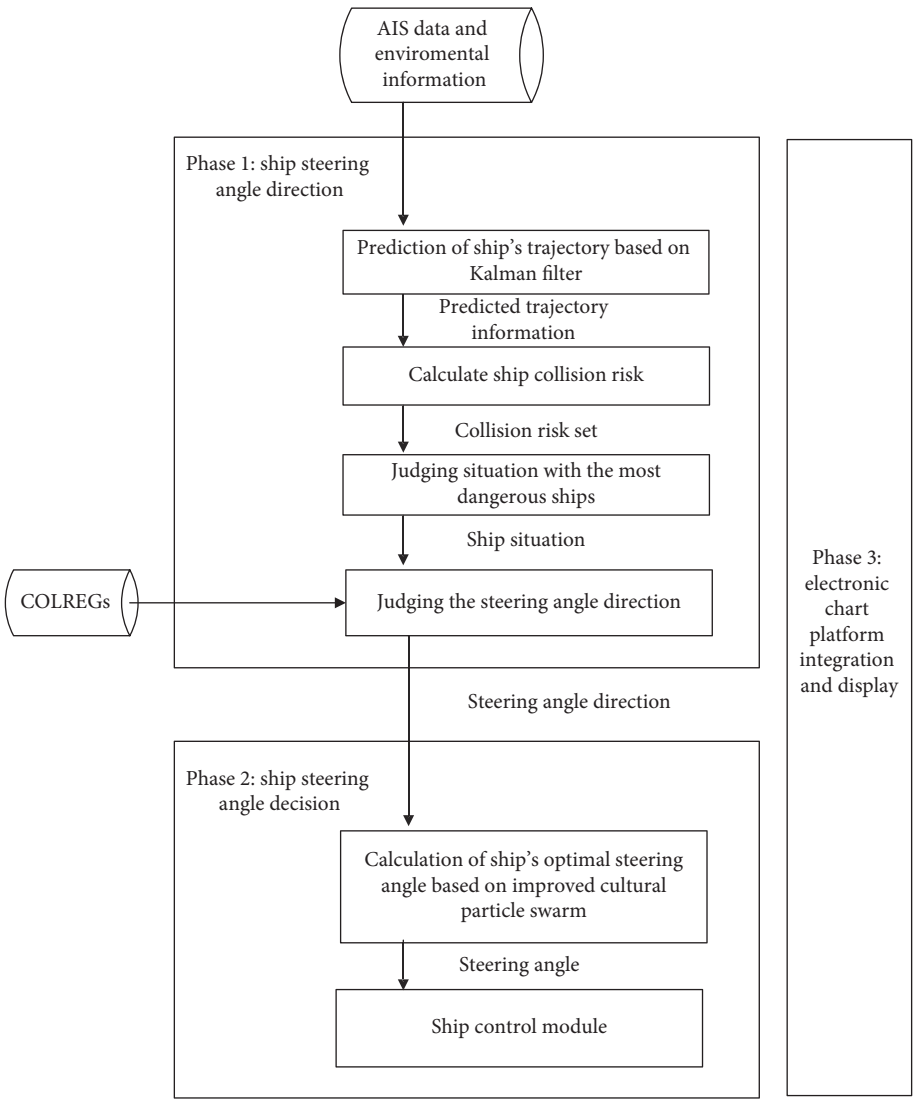


Figure 1: A ship collision avoidance decision framework based on cultural particle swarm.

information with collision risk in the navigation water area. At present, although a large number of AIS instruments have been installed in ship and shore-based management, due to various subjective and objective reasons, as well as congestion and network transmission of AIS ship stations and shore stations, such as improper AIS operation and data broadcast by AIS equipment due to ships, the distance from the AIS base station, the abnormal operation of the AIS device itself, or the artificial shutdown of the AIS device caused the AIS data received by the shore-based AIS to be incorrect and the data location report to be updated in a timely manner.

In order to compensate for the delayed data update caused by AIS data blockage and other reasons, resulting in inaccurate or large errors in the ship's trajectory, this paper uses the Kalman filter algorithm and considers the measurement noise caused by the AIS data transmission. Smoothing and predicting the ship's motion trajectory can reduce the error caused by the AIS data transmission, improve the accuracy of the data, and further improve the accuracy of collision avoidance decisions.

4.1.1. Basic Concepts

(1) *AIS Data*. AIS (automatic ship identification system) is a new navigation aid system for shore-to-ship, ship-to-shore, and ship-to-ship communication [29]. AIS equipment can send own ship information such as latitude and longitude, speed, heading, ship name, and other information to the target ship via VHF channel after processing by the processor and automatically obtain the position of other ships with similar equipment in real time, speed, heading, etc.

(2) *Mercator Projection*. Mercator projection is a map projection technology that uses an equiangular projection method. The principle is to assume that the Earth is enclosed in a hollow cylinder, whose datum of latitude (equator) is tangent to the cylinder, and then imagine a lamp in the center of the Earth, project the figure on the sphere onto the cylinder, and expand the cylinder to get a map drawn by the "Mercator projection" on the selected datum line [30].

The experimental model needs to convert the WGS84 latitude and longitude coordinates in the AIS data to Mercator x, y coordinates to predict the ship's trajectory. According to the principle of Mercator projection, the projection of the equator is X axis, the projection of the prime meridian is Y axis, and the projection of the intersection of the two is the origin of the coordinate, which is positive toward east and north and negative toward west and south, forming the Mercator plane rectangular coordinate system. Let the long axis of the Earth be a and the short axis be b. The longitude of a point on the Earth is $\theta \in (-\pi, +\pi)$ with latitude $\alpha \in (-(\pi/2), +(\pi/2))$; according to the equiangular principle, the coordinate conversion formula is as follows:

$$\begin{cases} x = a * \theta, \\ y = a * \ln\left(\tan\left(\frac{\pi}{4} + \frac{\alpha}{2}\right) + \frac{e}{2}\ln\left(\frac{1 - e * \sin\alpha}{1 + e * \sin\alpha}\right)\right), \end{cases} \tag{1}$$

where$e = \sqrt{a^2 - b^2}/a^2$ is the first eccentricity of the Earth's ellipsoid, and formula (1) (θ, α) translates to plane coordinates (x, y).

(3) *Trajectory Vector Set*. Modeling the two-dimensional plane X axis and Y axis of Euclidean space, using vectors in the directions of the two coordinate axes to represent trajectory data, and let T be the trajectory vector set. $T = \{Trj_1, Trj_2, \ldots, Trj_n\} = \{(p_x^1, p_y^1), (p_x^2, p_y^2), \ldots (p_x^n, p_y^n)\} = \{(p_{x,}^1 p_x^2, \ldots, p_x^n)^T, (p_y^1, p_y^2, \ldots, p_y^n)^T\}$, where $p_x^i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$represents the projection vector set of the ith trajectory in the X axis direction, $p_y^i = (y_{i1}, y_{i2}, \ldots, y_{id})^T$represents the projection vector set of the ith trajectory in the Y axis direction, and(p_x^i, p_y^i) is called the vector set of the trajectory Trj_i.

(4) *Kalman Filtering*. The core is to use recursive algorithms to estimate the optimal system state and realize the prediction of target motion behavior [31]. The Kalman filtering system of state equations and observation equations is as follows:

$$\begin{cases} X(k+1) = A(k)X(k) + \Gamma(k)U(k) + T(k)W(k), \\ Z(k) = H(k)X(k) + V(k), \end{cases} \tag{2}$$

where $X(k+1)$ represents $k+1$ state value at the moment, $A(k)$ is the state transition matrix, $\Gamma(k)$ is the control matrix, $U(k)$ is the control input vector, $T(k)$ is the interference transfer matrix, $W(k)$ is the system state noise of the motion model, $Z(k)$ is the observation vector, $H(k)$ is the observation matrix, and $V(k)$ is the observation noise.

(5) *Prediction Error*. For the geometric spatial error between the predicted trajectory point and the actual trajectory point, the root mean square error is shown as follows:

$$\text{RMSE} = \frac{\sum_{i=1}^{k} \sqrt{(x_i' - x_i)^2 + (y_i' - y_i)^2}}{k}, \tag{3}$$

where (x_i, y_i) represents the position of the actual trajectory point, (x_i', y_i') position information represents predicted trajectory points, and k represents the number of predicted trajectory points.

4.1.2. Ship Trajectory Prediction. In this paper, the AIS observation data transmitted by the VHF channel are used to optimally estimate the system state such as the ship position and speed to make real-time prediction of the ship trajectory.

(1) Ship Trajectory Modeling. The latitude and longitude of the ship are transformed into coordinates (x, y) in the Cartesian coordinate system through Mercator projection. $W_x(k)$ and $W_y(k)$ are the two orthogonal components of Gaussian white noise W(k) with zero mean and variance σ_W^2, $W_x(k)$ and $W_y(k)$ are independent of each other at any time. The values are related to factors such as actual sea wind, waves, and weather. $V_x(k)$ and $V_y(k)$ are two orthogonal components of Gaussian white noise V(k) with zero mean and variance σ_V^2, $V_x(k)$ and $V_y(k)$ are independent of each other at any time. The value is related to the error of the actual AIS information in the detection and transmission process. $W(k)$ and $V(k)$ are independent Gaussian white noises, which the corresponding covariances are Q and R.

The model obtains the optimal state estimate $X'(k)$ of the system at time k based on the number of observations transmitted by AIS in the first kt times. There are two different update processes in the process of the stochastic linear discrete Kalman filtering cycle, which are the time update process and the observation update process. The time update process is to predict the state at the current moment according to the optimal state estimation at the previous moment (see equation (4)). The observation update process is that after predicting the ship's trajectory point at $k+1$ time, the observed value of the ship's trajectory point at the actual $k+1$ time is used for linear fitting to find out the optimal estimated position of the trajectory point. That is to say, according to the observed value and the predicted value, the optimal estimation point of the ship's trajectory at $k+1$ time is derived through the observation update equation, as shown in equation (5).

$$\begin{cases} x(k+1, k) = x(k, k) + v(k)t\cos(\theta(k)), \\ y(k+1, k) = y(k, k) + v(k)t\sin(\theta(k)), \end{cases} \tag{4}$$

$$\begin{cases} x(k+1, k+1) = x(k+1, k) + K(k+1) * (x_z(k+1) - x(k+1, k)), \\ y(k+1, k+1) = y(k+1, k) + K(k+1) * (y_z(k+1) - y(k+1, k)), \end{cases} \tag{5}$$

where $x(k+1,k)$ is the predicted x coordinate of the ship at time $k+1$. $y(k+1,k)$ is the predicted y coordinate of the ship at time $k+1$. $x(k,k)$ is the optimal estimate of the x coordinate of the ship at time k. $y(k,k)$ is the optimal estimate of the y coordinate of the ship at time k. $v(k)$ is the ship's speed at time k, the $\theta(k)$ is the heading of the ship at time k, and t is the time interval. $K(k+1)$ is the filter gain matrix at time $k+1$. For the value and the update formula of the covariance $P(k+1,k+1)$ of the optimal state estimation at time, see formula (6).

$$\begin{cases} P(k+1,k) = A(k)P(k,k)A(k)^T + T(k)Q(k)T(k)^T, \\ S(k+1) = H(k+1)P(k+1,k)H(k+1)^T + R(k+1), \\ K(k+1) = P(k+1,k)H(k+1)^T S(k+1)^{-1}, \\ P(k+1,k+1) = P(k+1,k) - K(k+1)S(k+1)K(k+1)^T. \end{cases} \tag{6}$$

Where $Q(k)$ is the system noise of the ship at node k, $R(k)$ is the observed noise, $P(k,k)$ is the error variance matrix, $A(k)$ is the ship state transition matrix, $P(k+1,k)$ is the error variance matrix of the predictive state $x(k+1,k)$ and $y(k+1,k)$, and $K(k)$ is the filter gain matrix. Besides that, $W_x(k)$ and $W_y(k)$ have zero mean. Gaussian white noise with variance σ_W^2 and the time update equation of equation (5), we can get $T(k)$ and $A(k)$ as the identity matrix. Furthermore, the coordinates do not undergo unit conversion. The observation matrix $H(k)$ is also the identity matrix. According to the above formula, the optimal position prediction value of the ship at a single step is obtained. If the ship nt position is predicted, the ship trajectory prediction can be completed iteratively.

The ship trajectory prediction algorithm based on Kalman filtering is shown in Algorithm 1. The detailed steps are as follows.

In Algorithm 1, the function *trajectory prediction*(T) is used to collect and decode ship AIS data and perform (x, y) preprocessing operations such as coordinate transformation. The function initParameters is used to determine the parameters of the ship's motion model according to the state equation and observation equation of the system; besides, this function is also used to initialize the ship's A, U, and Q with R and other parameters. The function getCurrentState(D) is based on the optimal state estimate at the initial moment $X(0,0)$ and estimates error variance matrix $P(0,0)$, predicting the ship trajectory value at the next moment according to the system state equation $X(1,0)$ and obtaining the covariance matrix of the estimation error at the same time $P(1,0)$. According to the observed value $Z(1)$ of the ship trajectory at the next moment, theoptimal state estimation value $X(1,1)$ and the covariance matrix $P(1,1)$ of the optimal estimation error are obtained to complete the single-step filtering process of the ship trajectory. Then iteratively obtain the optimal state estimate $X(n-1,n-1)$ at the time of $n-1$ to complete the filtering process. The function of Kalman Predict(D) is to estimate the position of the ship's trajectory point at time n+1 based on the optimal state estimation $X(n-1,n-1)$ at time n-1 and the observed value of current time n. The function of getRMSE(p, p') is to compare the predicted point p' with real trajectory point p to calculate the filtering prediction error. Finally, repeat the operation k times to complete the prediction of the future k steps of ship trajectory points. Then, calculate and output the average prediction error.

(2) Simulation Experiment Verification. Experimental model using s_k indicates that the ship's real location is at the sampling time kt.$y(k)$ means observations of AIS data transmission at the moment kt, where $y(k) = s(k) + v(k)$, and $v(k)$ represents observation noise. $s'(k)$ means speed of ship at the moment kt, acceleration $a(k)$ is a combination of maneuvering acceleration $u(k)$ and random acceleration $w(k)$. The uniform acceleration motion and acceleration formula of the ship is as follows:

$$\begin{cases} s(k+1) = s(k) + s'(k)t + 0.5t^2 a(k), \\ s'(k+1) = s'(k) + ta(k), \\ a(k) = u(k) + w(k), \end{cases} \tag{7}$$

where $u(k)$ are the control signals of the ship's power system; define the system status $x(k)$ at the sampling time kt for the position and speed of the ship; the equations of motion and observation of the ship are shown as follows:

$$\begin{cases} \begin{bmatrix} s(k+1) \\ s'(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(k) \\ s'(k) \end{bmatrix} + \begin{bmatrix} 0.5T_0^2 \\ T_0 \end{bmatrix} u(k) + \begin{bmatrix} 0.5T_0^2 \\ T_0 \end{bmatrix} w(k), \\ y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s(k) \\ s'(k) \end{bmatrix} + v(k). \end{cases} \tag{8}$$

This experiment does not consider the ship's own dynamic factors $(u(k) = 0)$ and state vector X_k generalized to four dimensions; its expression and system equations are shown in equation (9), and the initial position is set as $(-100\,\text{m}, 200\,\text{m})$, X shaft speed is set as 2 m/s, and Y shaft speed is set as 20 m/s. The experimental record of the ship trajectory formed at 20 time points is shown in Figure 2, and the error analysis diagram is shown in Figure 3.

$$\begin{cases} \begin{bmatrix} x_k \\ x_k^- \\ y_k \\ y_k^- \end{bmatrix} = \begin{bmatrix} 1, T, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, T \\ 0, 0, 0, 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ x_{k-1}^- \\ y_{k-1} \\ y_{k-1}^- \end{bmatrix} + \begin{bmatrix} 0.5T^2, 0 \\ T, 0 \\ 0, 0.5T^2 \\ 0, T \end{bmatrix} w_k, \\ Z_k = \begin{bmatrix} 1, 0, 0, 0 \\ 0, 0, 1, 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_k^- \\ y_k \\ y_k^- \end{bmatrix} + v_k, \\ X_k = [x_k, x_k^-, y_k, y_k^-]. \end{cases} \tag{9}$$

The observed ship trajectory obviously oscillates in Figure 2, indicating that the measurement noise has a great impact, and after Kalman filtering, the filtering estimation is relatively close to the true trajectory of the ship. Figure 3 shows that the observed noise of the ship displacement is

```
Input: trajectory dynamic AIS dataset of moving ships T = {Trj_1, Trj_2, ..., Trj_n}
Output: mean value of ship trajectory error RMSE
D = trajectPretreatment (T)
initParameters ()
state = getCurrentState (D)
for i = 1 to k:
    p = KalmanPredict (D)
    e[i] = getRMSE (p, p')
end for
RMSE = (∑_{i=1}^{k} e[i])/k
Output RMSE
```

Algorithm 1: Ship trajectory prediction algorithm based on Kalman filter.

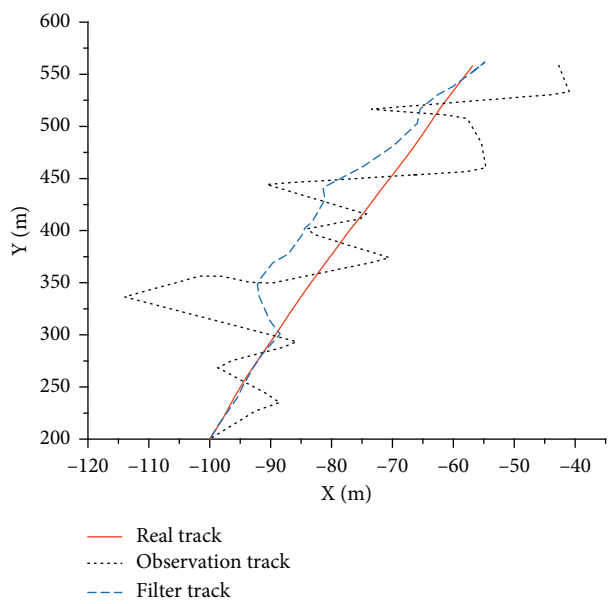


Figure 2: Ship tracking trajectory.

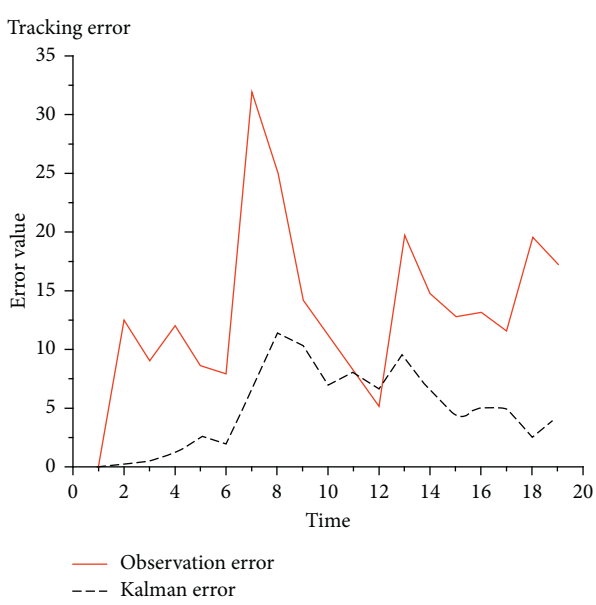


Figure 3: Ship tracking error.

close to 32 m, and the experiment is only simulation; the actual ship AIS measurement error will not be so large. After Kalman filter processing, the ship's displacement deviation is reduced to less than 12 m. Experiments show that Kalman filter can limit the impact of noise to the greatest extent and better predict the position after the ship.

4.2. Calculation of Ship Collision Risk. The membership function of different factors determines the value of ship collision risk. The collision risk of ships plays a key role in measuring the collision risk of ships and is an important basis for ship collision avoidance decision making [32]. The crew can decide the timing and order of avoidance according to the collision risk. There are many factors that affect the collision risk of ships. From the statistical research on the crew's collision avoidance behavior at sea by experts and scholars [33] and the research on the decision-making process, it can be seen that the main factors for judging the collision risk are intership distance, azimuth, DCPA, TCPA, and speed ratio. These factors can fully reflect the encounter

situation with the target, so the membership function of these factors is used as input to infer the collision risk of the ship.

4.2.1. Calculation of Influence Parameters. Kalman filtering prediction is used to obtain the ship X axis position coordinate x_0, the Y axis position coordinate y_0, the own ship's speed on the X axis component x_{v0} and the Yaxis component y_{v0}, other ship X axis position coordinate x_1, Y axis position coordinate y_1, the component x_{v1} on the X axis, and the component x_{v1} on the y_{v1} axis. Then, the longitude A_j, latitude A_w, heading A_h, and speed A_S of the own ship and the longitude B_j^m, latitude B_v^m, heading B_h^m, and speed B_S^m of other ships are obtained through the reverse solution of the Mercator projection equation. By substituting the above information into equation (10), the relevant influencing factor values can be solved.

$$\begin{cases} D = 2 * R * \arcsin\left(\sqrt{\sin^2\left(\dfrac{B_w^m - A_w}{2}\right) + \cos(A_w) * \cos(B_w^m) * \sin\left(\dfrac{B_j^m - A_j}{2}\right)}\right), \\ T_B = \begin{cases} \arctan\left(\dfrac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w}\right); & \left(\arctan\left(\dfrac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w}\right) \in \text{first quardrant}\right), \\ 360 + \arctan\left(\dfrac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w}\right); & \left(\arctan\left(\dfrac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w}\right) \in \text{second quardrant}\right), \\ 180 - \arctan\left(\dfrac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w}\right); & \left(\arctan\left(\dfrac{(B_j^m - A_j) * \cos(B_w^m)}{B_w^m - A_w}\right) \in \text{third and fourth quardrants}\right), \end{cases} \\ B = A_h - TB, \\ \mathrm{RSH} = \arctan\left(\dfrac{B_s^m * \cos(B_h^m) - A_s * \cos(A_h)}{B_s^m * \sin(B_h^m) - A_s * \sin(A_h)}\right), \\ \mathrm{RS} = \sqrt{B_s^{m^2} + A_s^2 - 2 * B_s^m * A_s * \cos(A_h - B_h^m)}, \\ \mathrm{DCPA} = D * \sin(\mathrm{RSH} - RA - 180^\circ), \\ \mathrm{TCPA} = D * \cos(\mathrm{RSH} - RA - 180^\circ), \end{cases} \tag{10}$$

where D, TB, B, RSH, and RS are the distance, azimuth, relative azimuth, relative speed course, and relative speed between ships.

4.2.2. Establishment of Collision Risk Model. The collision risk calculation method used in this paper is calculated through fuzzy sets. Fuzzy set is a set used to express the concept of fuzziness. It is defined as a given research range U; then a mapping from U to the unit interval [0, 1] is called a fuzzy set on U or a set of U fuzzy subset. Fuzzy set is described by membership function. The membership function occupies an extremely important position in fuzzy set theory. It is defined as if any element x in the research range U has a number $A(x) \in [0, 1]$ corresponding to it, then A is called the fuzzy set on U, and $A(x)$ is called the membership degree of x to A. When x changes in U, $A(x)$ is a function called A's membership function. In the fuzzy set, the value range of its characteristic function is [0, 1] interval; this characteristic function is also called membership function. Membership function is an important basis of fuzzy set theory, so how to determine the membership function is a key issue. However, the object is "fuzzy" and empirical, so it is unrealistic to find a unified calculation method for membership. The fuzzy distribution method currently used is used to construct the membership function of each influencing factor to realize the calculation of ship collision risk. The membership function formula ((11)–(17)) used in this paper refers to the relevant formula in reference [33].

(1) DCPA Membership Function t_{DCPA}. DCPA is the distance that the ship will meet recently. Statistics show that when the distance between the two ships is less than the distance between the two ships, the possibility of collision between the ships is extremely large. According to the empirical data of collision avoidance in open waters, when DCPA $\leq 0.6\,n$ mile, it is dangerous, $t_{\mathrm{VD(DCPA)}} = 1$; when $0.6\,n$ mile $<$ DCPA $\leq 1.8\,n$ mile, it is considered very dangerous, $t_{\mathrm{VD(DCPA)}}$; when $0.6\,n$ mile $<$ DCPA $\leq 1.2\,n$ mile, it is considered dangerous, $t_{D(\mathrm{DCPA})}$; when $1.0\,n$ mile $<$ DCPA $\leq 1.6\,n$ mile, it is considered fair, $t_{N(\mathrm{DCPA})}$; when $1.4\,n$ mile $<$ DCPA $\leq 2.5\,n$ mile, it is considered safer, $t_{S(\mathrm{DCPA})}$; when DCPA $> 2.5\,n$ mile, it is considered very safe, $t_{\mathrm{VS(DCPA)}}$. The membership function is shown in equation (11), and the corresponding function image is shown in Figure 4.

$$
\begin{aligned}
t_{\mathrm{VD(DCPA)}} &= \begin{cases} 1, & \mathrm{DCPA} \leq 0.6\,n\,\mathrm{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-0.7), & 0.6\,n\,\mathrm{mile} < \mathrm{DCPA} < 0.8\,n\,\mathrm{mile}, \\ 0, & 0.8\,n\,\mathrm{mile} \leq \mathrm{DCPA}, \end{cases} \\
t_{D(\mathrm{DCPA})} &= \begin{cases} 0, & \mathrm{DCPA} \leq 0.6\,n\,\mathrm{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.2}(x-0.7), & 0.6\,n\,\mathrm{mile} < \mathrm{DCPA} < 0.8\,n\,\mathrm{mile}, \\ 1, & 0.8\,n\,\mathrm{mile} \leq \mathrm{DCPA} \leq 1\,n\,\mathrm{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-1.1), & 1\,n\,\mathrm{mile} < \mathrm{DCPA} < 1.2\,n\,\mathrm{mile}, \\ 0, & 1.2\,n\,\mathrm{mile} \leq \mathrm{DCPA}, \end{cases} \\
t_{N(\mathrm{DCPA})} &= \begin{cases} 0, & \mathrm{DCPA} \leq 1\,n\,\mathrm{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.2}(x-1.1), & 1\,n\,\mathrm{mile} < \mathrm{DCPA} < 1.2\,n\,\mathrm{mile}, \\ 1, & 1.2\,n\,\mathrm{mile} \leq \mathrm{DCPA} \leq 1.4\,n\,\mathrm{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-1.5), & 1.4\,n\,\mathrm{mile} < \mathrm{DCPA} < 1.6\,n\,\mathrm{mile}, \\ 0, & 1.6\,n\,\mathrm{mile} \leq \mathrm{DCPA}, \end{cases} \\
t_{S(\mathrm{DCPA})} &= \begin{cases} 0, & \mathrm{DCPA} \leq 1.4\,n\,\mathrm{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.3}(x-1.55), & 1.4\,n\,\mathrm{mile} < \mathrm{DCPA} < 1.7\,n\,\mathrm{mile}, \\ 1, & 1.7\,n\,\mathrm{mile} \leq \mathrm{DCPA} \leq 2\,n\,\mathrm{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.5}(x-2.25), & 2\,n\,\mathrm{mile} < \mathrm{DCPA} < 2.5\,n\,\mathrm{mile}, \\ 0, & 2.5\,n\,\mathrm{mile} \leq \mathrm{DCPA}, \end{cases} \\
t_{\mathrm{VS(DCPA)}} &= \begin{cases} 0, & \mathrm{DCPA} \leq 2.0\,n\,\mathrm{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.5}(x-2.25), & 2.0\,n\,\mathrm{mile} < \mathrm{DCPA} < 2.5\,n\,\mathrm{mile}, \\ 1, & 2.5\,n\,\mathrm{mile} \leq \mathrm{DCPA}. \end{cases}
\end{aligned}
\tag{11}
$$

(2) TCPA Membership Function t_{TCPA}. TCPA is the recent meeting time of the ship, which reflects the urgency of the meeting situation between the two ships. Statistics and empirical data show that TCPA ≤ 6 min Time, $t_{\mathrm{VD(TCPA)}} = 1$; when 8 min ≤ TCPA ≤ 9 min Time, $t_{D(\mathrm{TCPA})} = 1$; when 11 min ≤ TCPA ≤ 12 min Time, $t_{N(\mathrm{TCPA})} = 1$; when 14 min ≤ TCPA ≤ 15 min Time, $t_{S(\mathrm{TCPA})} = 1$; when TCPA ≥ 17 minTime, $t_{VS(TCPA)} = 1$. The membership function is shown in equation (12), and the corresponding function image is shown in Figure 5.

$$
\begin{aligned}
t_{\mathrm{VD(TCPA)}} &= \begin{cases} 1, & \mathrm{TCAP} \le 6\,\mathrm{min}, \\ 0.5 - 0.5\sin\frac{\pi}{2}(x-7), & 6\,\mathrm{min} < \mathrm{TCPA} < 8\,\mathrm{min}, \\ 0, & 8\,\mathrm{min} \le \mathrm{TCPA}, \end{cases} \\
t_{D(\mathrm{TCPA})} &= \begin{cases} 0, & \mathrm{TCAP} \le 6\,\mathrm{min}, \\ 0.5 + 0.5\sin\frac{\pi}{2}(x-7), & 6\,\mathrm{min} < \mathrm{TCPA} < 8\,\mathrm{min}, \\ 1, & 8\,\mathrm{min} \le \mathrm{TCPA} \le 9\,\mathrm{min}, \\ 0.5 - 0.5\sin\frac{\pi}{2}(x-10), & 9\,\mathrm{min} < \mathrm{TCPA} < 11\,\mathrm{min}, \\ 0, & 11\,\mathrm{min} \le \mathrm{TCPA}, \end{cases} \\
t_{N(\mathrm{TCPA})} &= \begin{cases} 0, & \mathrm{TCAP} \le 9\,\mathrm{min}, \\ 0.5 + 0.5\sin\frac{\pi}{2}(x-10), & 9\,\mathrm{min} < \mathrm{TCPA} < 11\,\mathrm{min}, \\ 1, & 11\,\mathrm{min} \le \mathrm{TCPA} \le 12\,\mathrm{min}, \\ 0.5 - 0.5\sin\frac{\pi}{2}(x-13), & 12\,\mathrm{min} < \mathrm{TCPA} < 14\,\mathrm{min}, \\ 0, & 14\,\mathrm{min} \le \mathrm{TCPA}, \end{cases} \\
t_{S(\mathrm{TCPA})} &= \begin{cases} 0, & \mathrm{TCAP} \le 12\,\mathrm{min}, \\ 0.5 + 0.5\sin\frac{\pi}{2}(x-13), & 12\,\mathrm{min} < \mathrm{TCPA} < 14\,\mathrm{min}, \\ 1 & 14\,\mathrm{min} \le \mathrm{TCPA} \le 15\,\mathrm{min}, \\ 0.5 - 0.5\sin\frac{\pi}{2}(x-16), & 15\,\mathrm{min} < \mathrm{TCPA} < 17\,\mathrm{min}, \\ 0, & 17\,\mathrm{min} \le \mathrm{TCPA}, \end{cases} \\
t_{\mathrm{VS(TCPA)}} &= \begin{cases} 0, & \mathrm{TCAP} \le 15\,\mathrm{min} \\ 0.5 + 0.5\sin\frac{\pi}{2}(x-16), & 15\,\mathrm{min} < \mathrm{TCPA} < 17\,\mathrm{min}, \\ 1, & 17\,\mathrm{min} \le \mathrm{TCPA}. \end{cases}
\end{aligned}
\tag{12}
$$

(3) D Membership Function t_D. D It is the distance between ships; the smaller the value is, the closer the target ship is and the greater the collision risk of the ship is. Statistical empirical data show that when $D \le 1.5\,n$ mile, $t_{\mathrm{VD}(D)} = 1$; when $1.7\,n$ mile $\le D \le 2\,n$ mile, $t_{D(D)} = 1$; when $2.2\,n$ mile $< D < 2.5\,n$ mile, $t_{N(D)} = 1$; when

$2.7\,n\,\text{mile} < D < 3\,n\,\text{mile}$, $t_{S(D)} = 1$; when $D > = 3.2\,n\,\text{mile}$, $t_{\mathrm{VS}(D)} = 1$. The membership function is shown in equation (13), and the corresponding function image is shown in Figure 6.

$$
\begin{aligned}
t_{\mathrm{VD}(D)} &= \begin{cases} 1, & D \le 1.5\,n\,\text{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-1.6), & 1.5\,n\,\text{mile} < D < 1.7\,n\,\text{mile}, \\ 0, & 1.7\,n\,\text{mile} \le D, \end{cases} \\
t_{D(D)} &= \begin{cases} 0, & D \le 1.5\,n\,\text{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.2}(x-1.6), & 1.5\,n\,\text{mile} < D < 1.7\,n\,\text{mile}, \\ 1, & 1.7\,n\,\text{mile} \le D \le 2\,n\,\text{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-2.1), & 2\,n\,\text{mile} < D < 2.2\,n\,\text{mile}, \\ 0, & 2.2\,n\,\text{mile} \le D, \end{cases} \\
t_{N(D)} &= \begin{cases} 0, & D \le 2\,n\,\text{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.2}(x-2.1), & 2\,n\,\text{mile} < D < 2.2\,n\,\text{mile}, \\ 1, & 2.2\,n\,\text{mile} \le D \le 2.5\,n\,\text{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-2.6), & 2.5\,n\,\text{mile} < D < 2.7\,n\,\text{mile}, \\ 0, & 2.7\,n\,\text{mile} \le D, \end{cases} \\
t_{S(D)} &= \begin{cases} 0, & D \le 2.5\,n\,\text{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.2}(x-2.6), & 2.5\,n\,\text{mile} < D < 2.7\,n\,\text{mile}, \\ 1, & 2.7\,n\,\text{mile} \le D \le 3\,n\,\text{mile}, \\ 0.5 - 0.5\sin\dfrac{\pi}{0.2}(x-3.1), & 3\,n\,\text{mile} < D < 3.2\,n\,\text{mile}, \\ 0, & 3.2\,n\,\text{mile} \le D, \end{cases} \\
t_{\mathrm{VS}(D)} &= \begin{cases} 0, & D \le 3\,n\,\text{mile}, \\ 0.5 + 0.5\sin\dfrac{\pi}{0.2}(x-3.1), & 3\,n\,\text{mile} < D < 3.2\,n\,\text{mile}, \\ 1, & 3.2\,n\,\text{mile} \le D. \end{cases}
\end{aligned}
\tag{13}
$$

(4) B Membership Function t_B. B is the relative position between ships, and the danger of coming ships with different relative positions is different from their own ships. Statistical empirical data show that the safety of each area is ranked as E, A, B, C, D. E area arrival range is $(335^\circ \sim 5^\circ)$, and the meeting situation between the coming ship and own ship in this area is the most dangerous meeting situation. A area arrival range is $(5^\circ \sim 67.5^\circ)$, and the meeting situation between the coming ship and own ship in this area is a starboard small angle crossing meeting situation, which is more dangerous. B area arrival range is $(67.5^\circ \sim 122.5^\circ)$, and the meeting situation between the coming ship and own ship

in this area is a starboard high-angle crossing meeting situation, and the danger is average. C area arrival range is $(122.5^\circ \sim 210.5^\circ)$, and the meeting situation between the coming ship and own ship in this area is safer. D area arrival range is $(247.5^\circ \sim 355^\circ)$, and the meeting situation between the incoming ship and own ship in this area is safe. The membership function is shown in equation (14), and the corresponding function image is shown in Figure 7.

$$t_{\mathrm{VD}(B)} = \begin{cases} 1 - \dfrac{B}{360^\circ} & (0^\circ \le B \le 5^\circ), \\ \dfrac{B}{360^\circ} & (355^\circ \le B \le 0^\circ), \end{cases}$$
$$t_{D(B)} = 1 - \frac{|B - 36.25^\circ|}{360^\circ} \quad (5^\circ < B < 67.5^\circ),$$
$$t_{N(B)} = 1 - \frac{|B - 90^\circ|}{360^\circ} \quad (67.5^\circ \le B \le 122.5^\circ),$$
$$t_{S(B)} = 1 - \frac{|B - 180^\circ|}{360^\circ} \quad (122.5^\circ < B < 247.5^\circ),$$
$$t_{\mathrm{VS}(B)} = 1 - \frac{|B - 301.75^\circ|}{360^\circ} \quad (247.5^\circ \le B < 355^\circ). \tag{14}$$

(5) Ship Speed Ratio Membership Function t_K. Define the speed ratio K of the own ship and the targetship m as A_s/B_s^m. The statistical research on collision avoidance shows that when the own ship is a low-speed ship, collision avoidance actions must be taken as soon as possible and it needs to turn a large angle. Take $t_K = 0.5$ as the boundary point between safety and danger. The larger the value of t_K, the higher the risk of collision. The membership function is shown in equation (15), and the corresponding function image is shown in Figure 8.

$$t_K = \frac{1}{1 + (1/K)^2}. \tag{15}$$

(6) Calculation of Ship Collision Risk. Drawing on the research results at home and abroad and the experience of collision avoidance in open waters at sea, this paper establishes the evaluation criteria for the danger avoidance of each factor as shown in Table 1. The importance of the factor set {DCPA, TCPA, D, B, K}that affects the collision risk of ships is in order of DCPA > TCPA > D > B > K. According to the investigation in the literature [34], during the course of crew training bridge resource management, the corresponding weight of the factor set is $W = [0.4, 0.367, 0.167, 0.033, 0.033]$. The final risk vector R can be obtained through the evaluation matrix formed by the weight vector W and membership function of each factor (see equation (16)).

$$R = [W_{\mathrm{DCPA}}, W_{\mathrm{TCPA}}, W_D, W_B, W_K] \begin{bmatrix} t_{\mathrm{VD(DCPA)}} & t_{D(\mathrm{DCPA})} & t_{N(\mathrm{DCPA})} & t_{S(\mathrm{DCPA})} & t_{\mathrm{VS(DCPA)}} \\ t_{\mathrm{VD(TCPA)}} & t_{D(\mathrm{TCPA})} & t_{N(\mathrm{TCPA})} & t_{S(\mathrm{TCPA})} & t_{\mathrm{VS(TCPA)}} \\ t_{\mathrm{VD}(D)} & t_{D(D)} & t_{N(D)} & t_{S(D)} & t_{\mathrm{VS}(D)} \\ t_{\mathrm{VD}(B)} & t_{D(B)} & t_{N(B)} & t_{S(B)} & t_{\mathrm{VS}(B)} \\ t_{\mathrm{VD}(K)} & t_{D(K)} & t_{N(K)} & t_{S(K)} & t_{\mathrm{VS}(K)} \end{bmatrix}$$
$$= \begin{bmatrix} W_{\mathrm{DCPA}} t_{\mathrm{VD(DCPA)}} \oplus W_{\mathrm{TCPA}} t_{\mathrm{VD(TCPA)}} \oplus W_D t_{\mathrm{VD}(D)} \oplus W_B t_{\mathrm{VD}(B)} \oplus W_K t_{\mathrm{VD}(K)} \\ W_{\mathrm{DCPA}} t_{D(\mathrm{DCPA})} \oplus W_{\mathrm{TCPA}} t_{D(\mathrm{TCPA})} \oplus W_D t_{D(D)} \oplus W_B t_{D(B)} \oplus W_K t_{D(K)} \\ W_{\mathrm{DCPA}} t_{N(\mathrm{DCPA})} \oplus W_{\mathrm{TCPA}} t_{N(\mathrm{TCPA})} \oplus W_D t_{N(D)} \oplus W_B t_{N(B)} \oplus W_K t_{N(K)} \\ W_{\mathrm{DCPA}} t_{S(\mathrm{DCPA})} \oplus W_{\mathrm{TCPA}} t_{S(\mathrm{TCPA})} \oplus W_D t_{S(D)} \oplus W_B t_{S(B)} \oplus W_K t_{S(K)} \\ W_{\mathrm{DCPA}} t_{\mathrm{VS(DCPA)}} \oplus W_{\mathrm{TCPA}} t_{\mathrm{VS(TCPA)}} \oplus W_D t_{\mathrm{VS}(D)} \oplus W_B t_{\mathrm{VS}(B)} \oplus W_K t_{\mathrm{VS}(K)} \end{bmatrix}. \tag{16}$$

Here, $\oplus$ is a hazard synthesis operator. According to the actual operation requirements of collision avoidance at sea, the synthesis operator is regulated as follows:

$$\alpha \oplus \beta = \min(1, \alpha + \beta). \tag{17}$$

4.2.3. Calculation of Ship Collision Risk. Literature [34] sets the size of the collision risk of the target ship relative to the own ship as a set $[-2, -1, 0, 1, 2]$. It is divided into five dangers: low danger, low danger, normal danger, high danger, and high danger, calculated by weighting the vector R by its weight value. Table 2 shows the relevant data and the influencing factors (the first is the own ship data) of the own ship and the 4 experimental ships at that moment. Figure 4 shows the hazard curves of the ship and the four experimental ships when the steering angle of the ship is limited within the range $[-35, 35]$. The horizontal axis is the number of steering angles of the ship. Negative values represent counterclockwise steering.

From Figure 9 and Table 2, it can be seen that when own ship is at the initial heading (the horizontal axis of the image is 0), the danger is first ship > second ship > third ship > fourth ship, and as the ship's clockwise steering angle

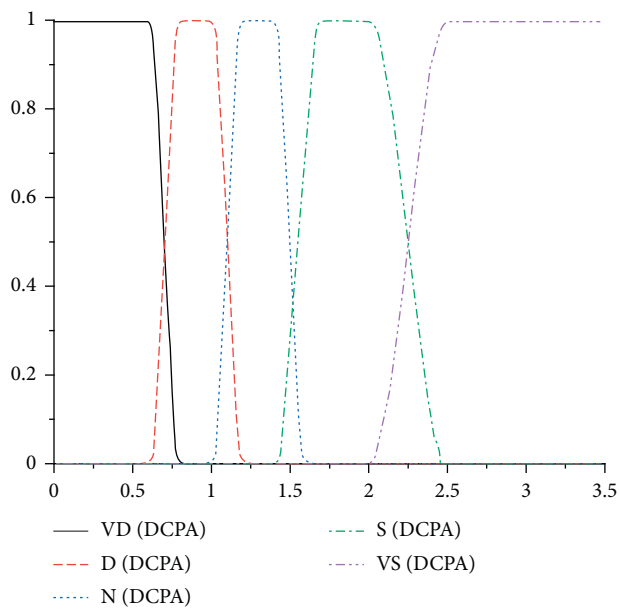


Figure 4: DCPA membership function image.

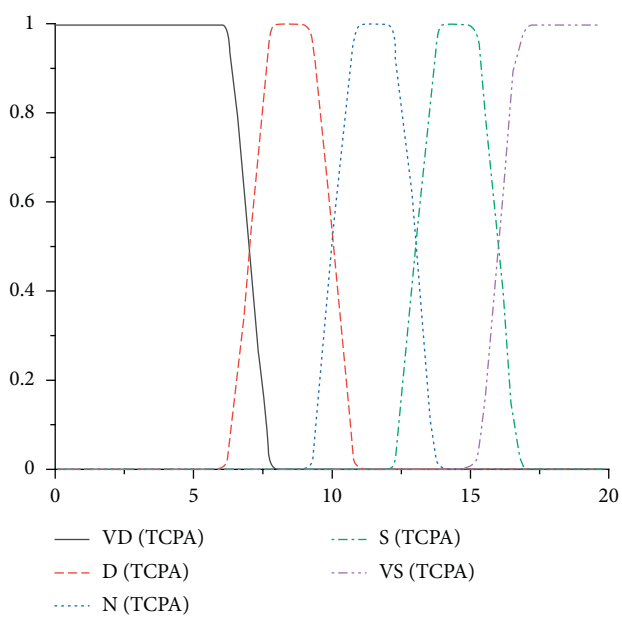


Figure 5: TCPA membership function image.

increases, the danger of the second ship becomes higher, making it more prone to collision risks.

4.3. Ship Steering Angle Decision. By analyzing the collision risk curve of the ship in Figure 4 and following the rule that the ship will always avoid the most dangerous ship each time when avoiding a collision, the lower the collision risk of the ship in the curve, the more likely it is to collide. We analyze the collision risk curve of the ship, take the lowest value of the ship's danger degree corresponding to each steering angle, and record the name (serial number) of the ship with

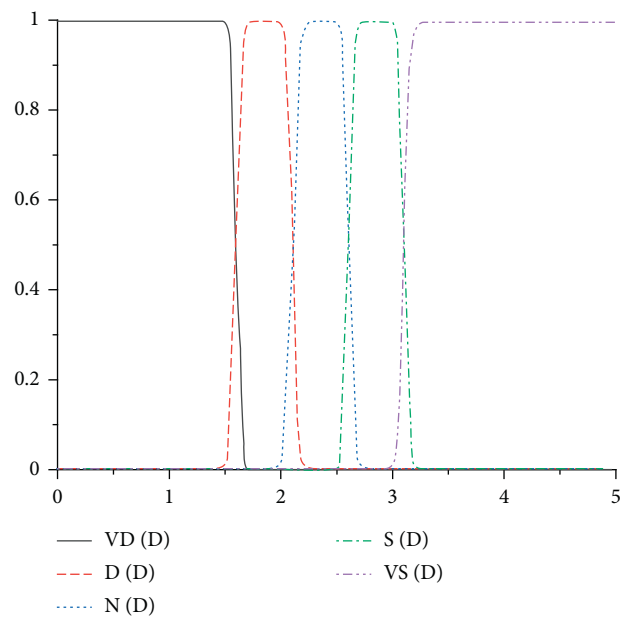


Figure 6: *D* membership function image.

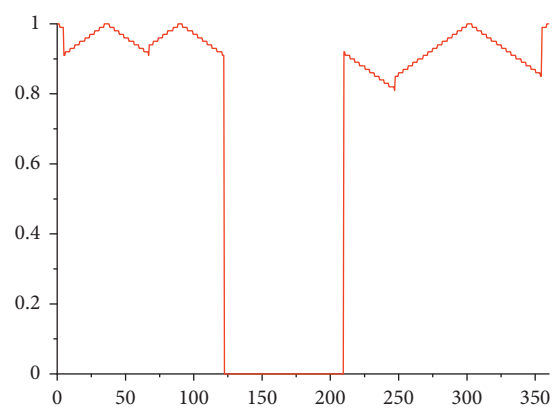


Figure 7: *B* membership function image.

the lowest collision danger when the steering angle is zero so as to analyze the current situation of the ship corresponding to the most dangerous target ship situation and determine the direction of the ship's steering angle.

The encounter situation of ships is determined in accordance with the applicable provisions of the rules for avoiding collisions. The rules for avoiding collisions are the basic guidelines for the safe navigation of ships at sea, and they are also an important legal basis for regulating ship avoidance actions. The rules clearly stipulate the methods of avoidance of ships in various situations and play an important role in guiding the safe navigation and avoidance of ships. The encounter situation is mainly divided into two types: first, under the provisions of the collision avoidance rules, the meeting situation is divided according to the possible collision avoidance actions of the crew and the degree of collision avoidance difficulty. The second is from the angle of collision avoidance action (left and right collision avoidance) and

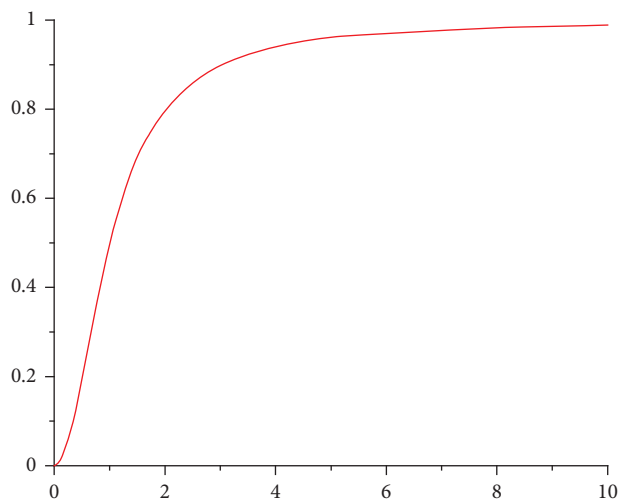


Figure 8: Ship speed ratio membership function image.

Table 1: Criteria for risk evaluation of various factors.

Factor	Low risk	Low risk	Average risk	High risk	High risk
DCPA (*n*mile)	2.5 ~ 4.0	1.2 ~ 2.5	1.0 ~ 1.6	0.6 ~ 1.2	0.6 ~ 0.8
TCPA (min)	18	15	12	9	6
D (*n*mile)	4.0	3.0	2.5	2.0	1.5
B (°)	301.75°	180°	90°	36.25°	0°
K	< 0.5	0.5 ~ 0.8	0.8 ~ 1.2	1.2 ~ 2	> 2

Table 2: Related data of own ship and other ships.

Longitude	Latitude	Course	Speed	Bearing	Relative bearing	Distance	Relative speed heading	Relative	DCPA	TCPA	Collision risk
23	30	0	50	—	—	—	—	—	—	—	—
23.07	30.02	220	60	71.72	71.72	4.33	201.89	103.42	−3.31	1.62	−0.077
22.88	30.01	310	60	275.52	275.52	7.23	256.03	47.36	2.41	8.63	0.057
23.05	29.98	260	60	114.78	114.78	3.18	224.36	84.51	−2.99	0.75	0.28
22.87	29.97	40	60	255.11	255.11	7.97	95.98	38.78	2.84	11.52	1.17

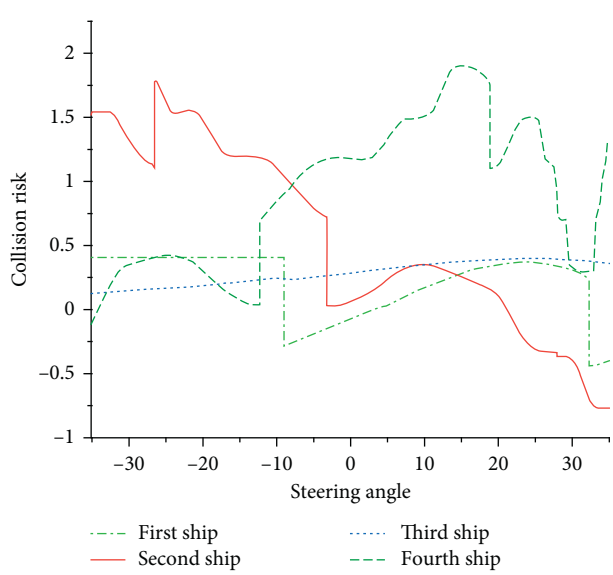


Figure 9: Curve of ship collision risk.

according to the relative azimuth (side angle) of the target ship [34]. Reference [28] divides three kinds of ship encounter situations, including 16 kinds of ship encounter situations, based on the relative position relationship between the own ship and the target ship, the avoidance responsibility, and the steering avoidance action method and good visibility. In this paper, by collecting the actual control experience of the master and crew and combining with the provisions of the International Regulations for Preventing Collisions at Sea in 1972, the specific directions of the ship's steering angles for the overtaking ship, the giving way ship, and the ship in the encounter are given.

Rule 13 of the Collision Avoidance Rule: "when a ship is catching up with other ships from a certain direction that is more than 22.5 degrees behind the other ship, that is, the position of the ship it is chasing, it can only see the taillight of the overtaken ship at night, but not the other. Any of its side lights should be considered to be overtaking." According to the rules of collision avoidance and actual navigational experience, when the ship is an overtaking

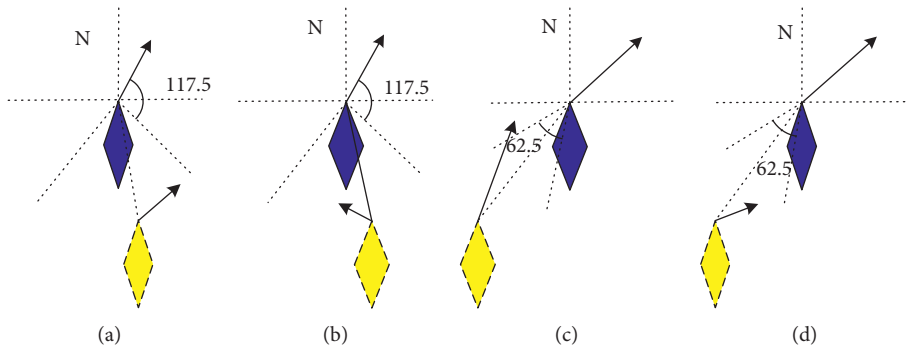


Figure 10: Overtaking situation (own ship is a direct ship).

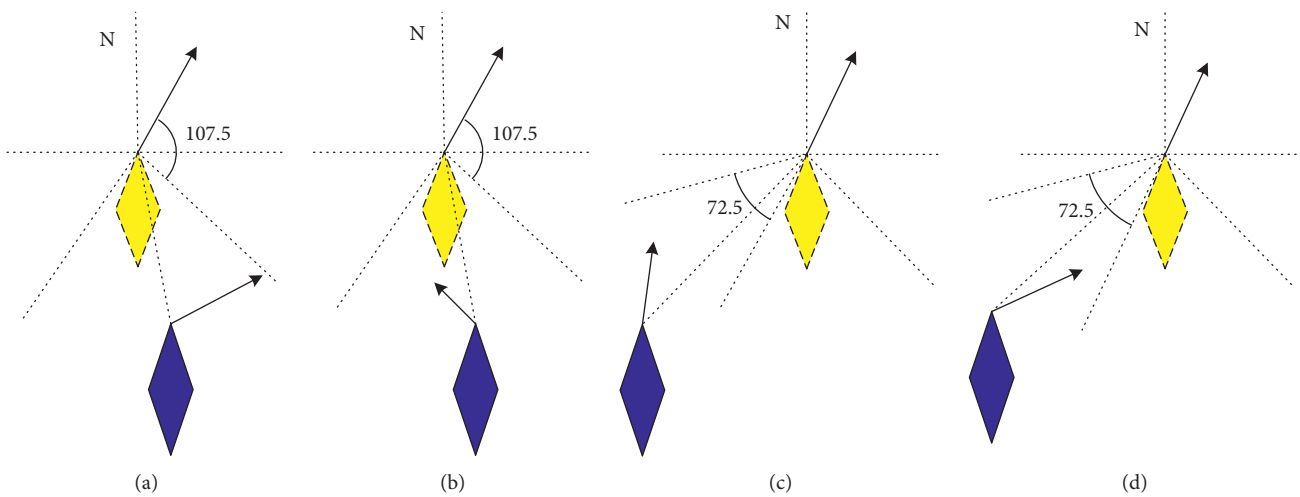


Figure 11: Overtaking situation (own ship is overtaking ship).

ship, the crew usually practices the rudder to the left. Regardless of any change in the orientation between the two ships, until the responsibility of the overtaken ship is relieved and the ship passes by to clear it, the situation is shown in Figures 10 and 11, where Figure 10 is a direct ship and Figure 11 is an overpass. In the following picture, the blue solid line is the main ship and the yellow dotted line is the target ship.

Rule 14 of the Collision Avoidance Rule: "when two motor ships meet in opposite or near opposite headings and there is a danger of collision, they each turn to the right so that they each pass through the port side of other ships"; according to collision avoidance, the crew members should normally steer the rudder to the right, and the situation is shown in Figure 12.

Rule 15 of the Collision Avoidance Rule: "when two motor boats cross each other and there is a danger of collision, a ship with other ships on the starboard side of the ship should make way for other ships. If the environment permits at the time, they should also avoid crossing the front of other ships." According to the rules of collision avoidance and actual navigational experience, when the ship is a yielding ship, the crew usually practices the rudder to the right until the responsibility of the yielding ship is relieved. The situation is shown in Figures 13 and 14. Among them, the first two small pictures in Figures 13 and 14 limit the speed ratio of own ship to the target ship.

According to the literature [35], the ship's danger is divided into 5 levels. The minimum value of the collision danger corresponding to the ship at the current moment is obtained through the curve in Figure 5. When the value is less than −0.4, the collision risk of the ship is more likely to require a collision avoidance decision. At the moment, the meeting situation of the ship with the lowest collision risk is determined. According to the above conclusions, the determination of the ship rudder angle direction is completed.

5. Decision Method of Optimal Steering Angle Based on Improved Cultural Particle Swarm

After finding the direction of ship's steering angle under the collision avoidance rules in Section 3, this section will find the optimal steering angle of the ship under this direction to achieve ship collision avoidance. In order to be more in line with the actual sailing environment of marine vessels and obtain more accurate results, this paper uses the CPSO (cultural particle swarm optimization) algorithm to establish

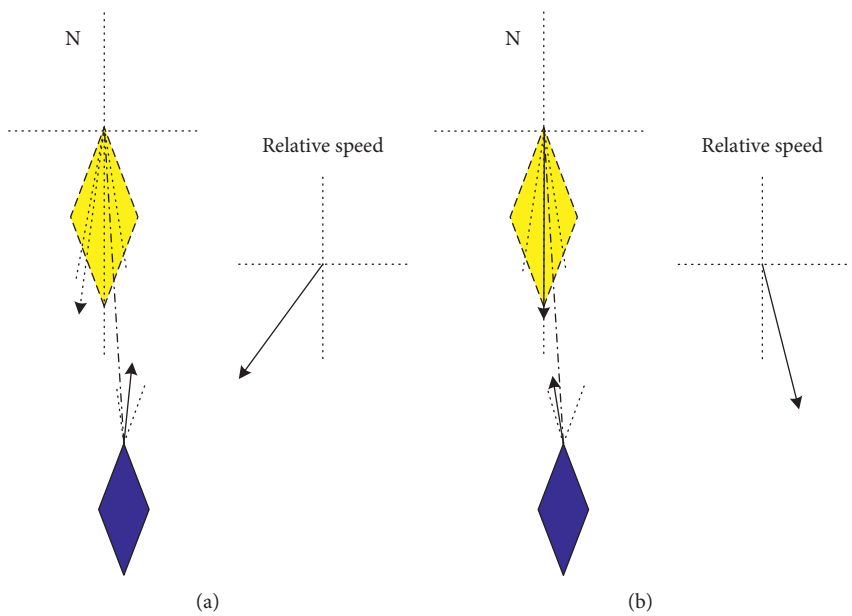


Figure 12: Meeting situation.

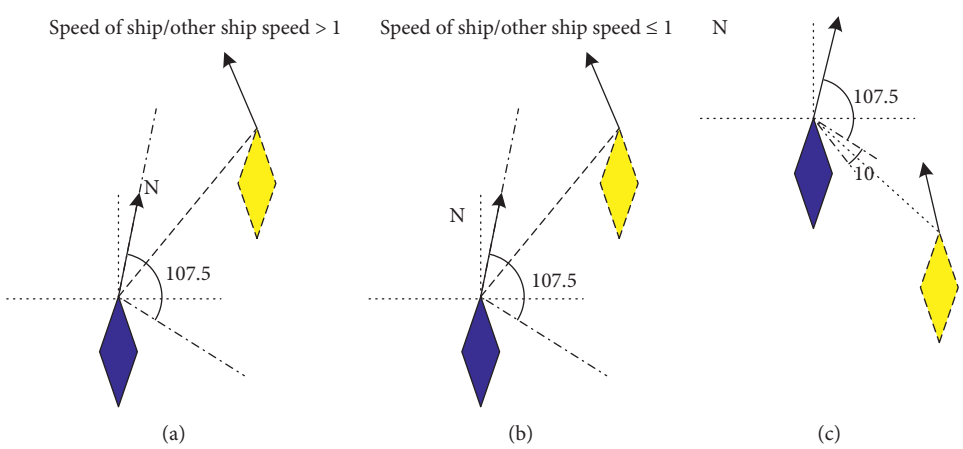


Figure 13: Crossing situation (own ship is a give way ship).

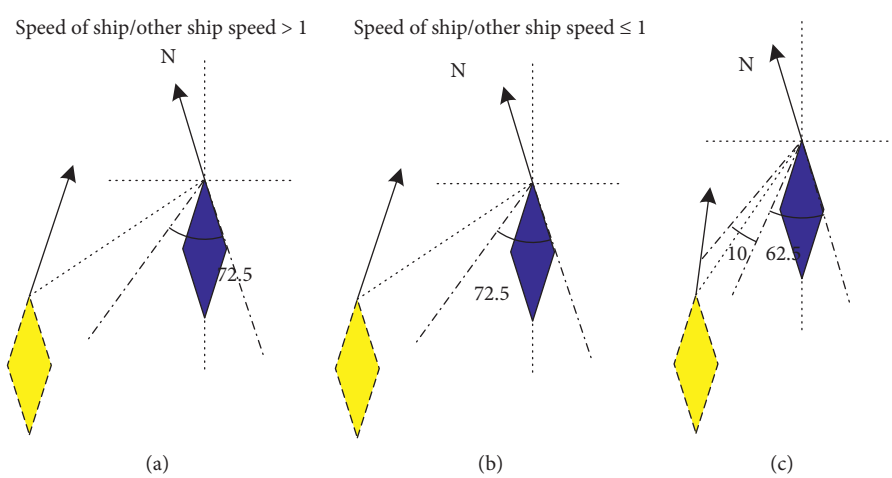


Figure 14: Cross situation (own ship is a direct ship).

the main swarm space (lower space) and knowledge space (upper space) of the particle swarm and improve the evolution efficiency through the interaction between the two levels so as to find the optimal steering angle of the ship more effectively.

5.1. Cultural Particle Swarm Algorithm

5.1.1. Cultural Algorithm. Cultural algorithm (CA) is a new global optimization search algorithm created by Reynolds et al. to solve complex calculations in 1994. Researchers [36] see culture as a carrier of information that potentially affects all members of society and is useful in guiding the evolutionary process of contemporaries and offspring. Cultural algorithm is a two-layer evolution system based on knowledge which contains two evolution spaces: one is a belief space composed of experience and knowledge acquired during the evolution process and the other one is the population space composed of specific individuals, which is solved iteratively through evolutionary operations and performance evaluation. The basic framework of the cultural algorithm is shown in Figure 15.

As shown in Figure 15, population space and belief space are two relatively independent evolutionary processes. The two spaces are linked by a set of communication protocols consisting of acceptor function Accept and influence function Influence. During the evolution process of individuals in the population space, individual experiences are formed, and the individual experiences are transferred to the belief space through Accept functions. Belief space compares and optimizes individual experience according to certain behavior rules to form population experience. Belief space is updated with Update functions based on existing population experience and new individual experience; the Influence function can use the empirical knowledge of the problem to be solved in the belief space to guide the evolution of the population space so that the population space can obtain higher evolution efficiency. In a cultural system, there are many kinds of knowledge, among which normative knowledge and situational knowledge are regarded as the most important knowledge. Normative knowledge provides behavioral norms and guiding principles for individuals, and situational knowledge provides examples for individuals. They can provide guidance information for the evolution of populations. objective function is the objective function (fitness function). Its function is to evaluate the fitness value of individuals in the population space. generate function generates the next generation of individuals based on the rules of individual behavior and the parameters of the parents. The select function selects a part of the newly generated individuals as the parents of the next generation of individuals according to the rules (see Algorithm 2).

5.1.2. Particle Swarm Algorithm. The term "group intelligence" was first coined by Beni et al. [37]. It was proposed in the research of cellular robot systems. Typical swarm intelligence consists of a group of simple agents with local influence and their environment. Although there is no centralized control structure in the swarm intelligence system to guide the behavior of these agents, local influences can often generate global behavior. Particle swarm optimization, as a typical swarm intelligence optimization algorithm, originates from the research on the simplified social model of bird swarms and the simulation of behaviors [38].

Suppose each particle in the particle swarm algorithm is flying at a certain speed in the n-dimensional search space; $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ is particles i's position, $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ is particles i's speed, and $p\text{best}_i = (p\text{best}_{i1}, p\text{best}_{i2}, \ldots, p\text{best}_{in})$ is optimal position experienced by particles i. The current optimal position, global optimal position, particle moving speed, and position update formula are shown in the following equation:

$$\begin{cases} p\text{best}_i(t+1) = \begin{cases} p\text{best}_i(t), & \text{if } f(X_i(t+1)) \geq f(p\text{best}_i(t)), \\ X_i(t+1), & \text{if } f(X_i(t+1)) < f(p\text{best}_i(t)), \end{cases} \\ g\text{best}(t) = \min\{f(p\text{best}_1(t)), f(p\text{best}_2(t)), \ldots, f(p\text{best}_N(t)), \\ v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1 \left(p\text{best}_{ij}(t) - x_{ij}(t)\right) + c_2 r_2 \left(g\text{best}_j(t) - x_{ij}(t)\right), \\ x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \end{cases} \tag{18}$$

where $g\text{best}(t)$ is the global optimal position, w is the inertia weight, i is the first i particles, j is particle dimension, c_1, c_2 are the acceleration factors, and r_1, r_2 are random numbers among $[0, 1]$. Particle swarm algorithm speed update equation consists of three parts: the first part is the previous speed of the particle; the second part is called the "cognitive" part, which comes from the experience and thinking of the particle itself; the acceleration factor c_1 can adjust the flight step size of the particle's best position; the third part is the "social" part of the particle, indicating information sharing and interaction between particles and acceleration factors c_2. You can adjust the particle's flight step to the optimal position of the group.

5.1.3. Cultural Particle Swarm Algorithm. The CA algorithm framework provides a computational model of multiple evolutionary processes, so from the perspective of the computational model, any evolutionary algorithm that meets the requirements of a cultural algorithm can be embedded in the cultural algorithm framework as an evolutionary process of population space. The current research studies mainly focus on the group intelligence algorithms such as GA, POS, and ACO. Unlike the existing evolutionary algorithms, CA provides a dual evolution mechanism. Based on the group space evolutionary algorithm, through the acquisition, preservation, analysis, and integration of the empirical knowledge generated by individual evolution, the evolution process of the population space is further guided so that the evolution speed of the population surpasses the evolution speed solely relying on ordinary evolution algorithms and has good global optimization performance.

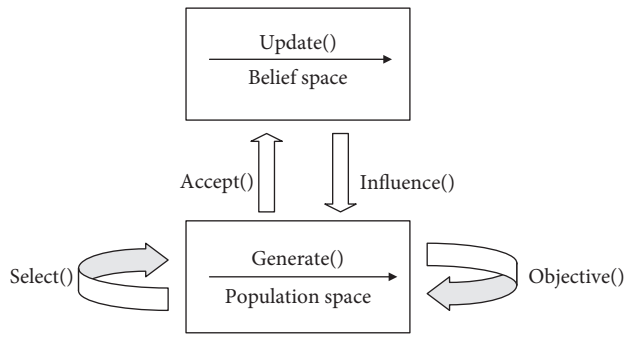


Figure 15: Graphical representation of the cultural algorithm.

```
begin
t = 0;
  POP(t); //Initialize the population space
  BLF(t); //Initialize the belief space
  repeat
  POP(t); //Assessing population space
    update (BLF(t), accept (POP(t)));
    generate (POP(t), influence (BLF(t)));
    t = t + 1;
    select POP(t) from POP(t−1);
    until Termination condition
  end
```

Algorithm 2: Cultural algorithm pseudocode.

The essence of cultural particle swarm optimization algorithm is a combination of cultural algorithm model and particle swarm optimization. During the PSO evolution process, particles track two "extreme values": one is the individual historical optimal solution, the other is the global historical optimal solution, and the global historical optimal solution is passed from generation to generation, which naturally forms the trajectory of the global optimal solution. These optimal solutions are regarded as the global knowledge information of iterative search which is kept continuously, and the knowledge space is updated; the knowledge solution is evolved. At the same time, the further solution of PSO group is guided. Through the dual evolution and mutual influence of PSO space (main group space) and knowledge space, the solution has better global search ability.

5.2. Improvements to the Cultural Particle Swarm Algorithm

5.2.1. Premature Convergence Judgment. The diversity of the particle swarm population is a prerequisite for the effective operation of the algorithm, and the greater the population diversity, the more likely it is to produce better offspring individuals. However, due to the solution process of the particle swarm algorithm, the diversity will be gradually lost, and the individuals will converge with each other. The particle with the optimal current position will guide other particles to approach it quickly. If the position of the current optimal particle is a local optimal solution, the entire group cannot search again and can fall into a precocious convergence state. Generally, the fitness of different distributed particles in the population space is different. The degree of diversity of particles can be expressed by the degree of dispersion of fitness distribution of particles. When the PSO algorithm is in the state of premature convergence or global convergence, particles will gather in one place or in several specific places. Therefore, the state of the particle swarm can be judged by studying the change of the particle fitness value in the population space, thereby determining whether the algorithm converges.

This paper introduces the index proposed in [34] to evaluate the premature convergence of the particle swarm, which is defined as: Let the size of the particle swarm be n. In the k-th iteration, the fitness value of the particle is f_i^k and the average fitness value of the particle swarm is f_{avg}^k; $\overline{f_{\text{avg}}^k}$ is obtained by averaging the fitness values of particles whose fitness value is better than f_{avg}^k, which is defined as follows:

$$\rho = \left| \frac{f_m^k - \overline{f_{\text{avg}}^k}}{f_m^{k-1} - \overline{f_{\text{avg}}^{k-1}}} \% \right|, \tag{19}$$

where ρ evaluates the degree of convergence of the particle swarm algorithm; the smaller ρ is, the more the particle swarm tends to converge.

5.2.2. Dynamic Adjustment of Inertia Weight. The standard particle swarm algorithm uses linearly decreasing inertial weights because the linearly decreasing inertial weight changes are too singular, and the process of searching the optimal steering angle of a ship by the particle swarm algorithm is very complicated and nonlinear; it is suitable for complex problems. Both ability and regulation ability are very limited. Aiming at the above problems, this paper proposes a strategy for dynamically adjusting the inertia weights, combining the changes of the inertia weights with indicators to evaluate the degree of premature convergence of the particle swarm:

$$w_i(t) = 1 - \left(1 + e^{-\rho}\right)^{-1}. \tag{20}$$

It can be seen from the above formula that the particle's inertia weight value changes within (0, 1), $w_i(t)$ describes the effect of the inertia of the i-th particle on the t-th velocity, and the value can adjust the global optimization ability of the PSO and local optimization capabilities. When the PSO is in a state of convergence, a large inertia weight is required to increase the search step size, thereby enhancing the global optimization capability of the PSO. When the PSO is in the global search state, a small inertia weight is required to reduce the search step size, thereby enhancing the local optimization capability of the PSO. When the index for evaluating the degree of premature convergence of the particle swarm decreases, $w_i(t)$ increases accordingly, which enhances the global search ability; when ρ increases, the value of $w_i(t)$ decreases accordingly, increasing the local search ability.

5.3. Optimal Steering Angle Decision Algorithm

5.3.1. Particle Swarm Space Design and Iteration Rules

(1) *Coding Scheme.* The particle swarm in the population space encodes the particle swarm according to the Michigan coding scheme, that is, each particle represents a ship collision avoidance strategy, and all the collision avoidance strategies correspond to a certain particle swarm, the particles are composed of different dimensions, and the dimensions of the particles are correspondingly related to the feature term; this article only searches for the ship's collision avoidance steering angle and does not involve other feature items such as ship speed. Therefore, the steering angle is expressed as coordinates x_i, $i = 1, 2, \ldots, N$; then, this particle swarm individual is encoded as $x_1 x_2 x_3 \ldots x_N$.

(2) *Fitness Function.* Determine the fitness function of the individual particle swarm based on the collision risk of the ship and set the number of ships that need to be avoided at present as M; the ship collision risk function is CollisionRisk, and the value of the fitness function is mainly related to the collision risk of the currently most dangerous ships, so its fitness function is expressed as

$$\text{objective}(\theta) = \text{Min}(\text{CollisionRisk}(i, \theta)), \quad i \in M. \tag{21}$$

Among them, θ represents the size of the steering angle, positive represents the clockwise direction, and negative represents the counterclockwise direction. According to the collision risk curve of the ship in Figure 9 of this paper, the fitness function curve at that point in time is obtained, as shown in Figure 16.

5.3.2. Design and Update Rules of Belief Space. The structure of the belief space uses the structure pair $\langle S, N \rangle$ proposed in article [39], where $S = \{s_1^t, s_2^t, \ldots, s_m^t\}$ is situational knowledge, which represents the optimal individual set, s_i^t represents the i-th optimal individual in the t generation population, and m represents the scale of the optimal individual set. This article only takes the current best individual s^t to update the situation knowledge S in belief space, which is $S = \{s^t\}$, and searching for the optimal steering angle of the ship is to find the maximum value $\text{Max}(\text{fitness}(\theta))$ of the fitness function, so the update formula is as follows:

$$s^{t+1} = \begin{cases} x_{\text{best}}^t, & \text{objective}\left(x_{\text{best}}^t\right) > \text{objective}\left(s^t\right), \\ s^t, & \text{otherwise}. \end{cases} \tag{22}$$

Among them, x_{best}^t represents the optimal particle in the t generation. In the current population, if the optimal particle x_{best} is better than the existing one s^t, then the knowledge of the belief space situation is adjusted and updated by replacing x_{best} and s^t.

$N = < X_1, X_2, \ldots, X_n >$ is standardized knowledge; it represents the value interval information of each variable. n is the number of variables, and X_i is $\langle I, L, U \rangle$, where I is defined as

$$I = [l, u] = \{x \mid l \le x \le u\}. \tag{23}$$

Among them, l and u represent the lower and upper boundaries of the particle steering angle decision, respectively, and L and U represent the fitness values corresponding to these two boundaries. In t-th iterations of the particle swarm, it is assumed that the steering angle parameter of i-th particle affects its lower boundary l_i^t and its fitness value L_i^t. Similarly, the steering angle parameter of the k-th particle affects its upper boundary u_k^t and its fitness value U_k^t. For the parameter update, see equation (24).

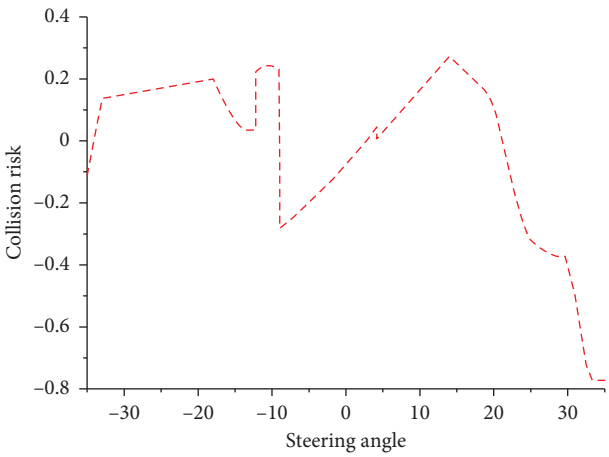


Figure 16: Fitness function curve.

$$l_i^{t+1} = \begin{cases} x_i^t, & x_i^t \le l_i^t \text{ or objective}\left(x_i^t\right) > L_i^t, \\ l_i^t, & \text{otherwise}, \end{cases}$$
$$L_i^{t+1} = \begin{cases} \text{objective}\left(x_i\right), & x_i^t \le l_i^t \text{ or objective}\left(x_i^t\right) > L_i^t, \\ L_i^t, & \text{otherwise}, \end{cases}$$
$$u_k^{t+1} = \begin{cases} x_k^t, & x_k^t \ge u_k^t \text{ or objective}\left(x_k^t\right) > U_k^t, \\ u_k^t, & \text{otherwise}, \end{cases}$$
$$U_k^{t+1} = \begin{cases} \text{objective}\left(x_k^t\right), & x_k^t \ge u_k^t \text{ or objective}\left(x_k^t\right) > U_k^t, \\ U_k^t, & \text{otherwise}. \end{cases} \tag{24}$$

5.3.3. Acceptance Function and Influence Function. This article uses a dynamic acceptance function, which introduces evolutionary algebra as a dynamic factor and adjusts the acceptance ratio. The expression is

$$\eta_{\text{accept}} = \left(p\% + \frac{p\%}{t}\right) * n. \tag{25}$$

Among them, n is the size of the population, t is the current evolutionary algebra, and $p\%$ is a predetermined fixed ratio. The belief space can affect the mutation operator in the population space in two ways: firstly, changing the step size of the variable change, and secondly, changing the direction of the variable. In this section, normative knowledge is used to adjust the variable step size, and situational knowledge is used to adjust its forward direction. The specific expression is as follows:

$$x_i^{t+1} = \begin{cases} x_i^t + \left|\text{size}\left(I_i^t\right) * N(-1,1)\right|, & x_i^t < s_i^t, \\ x_i^t - \left|\text{size}\left(I_i^t\right) * N(-1,1)\right|, & x_i^t > s_i^t, \\ x_i^t + \text{size}\left(I_i^t\right) * N(-1,1), & x_i^t = s_i^t, \end{cases} \tag{26}$$

where $N(-1,1)$ is a random number that follows a normal distribution and size(I_i^t) is the difference between the upper and lower bounds of the normative knowledge of the particle swarm at the t iteration. Through normative knowledge and situational knowledge, offspring individuals move as far as possible to a given interval (see Algorithm 3).

5.4. Experimental Verification. The experiment uses the data of own ship and related ships in Table 2 of Section 4, taking the total number of particles $n = 30$, the maximum number of iterations $I = 1000$, acceleration factor $c_1 = c_2 = 2$, ship steering angle whose range is [−35,35], the fitness function which is objective(θ) = Min(CollisionRisk(i, θ)), where i is the number of vessels to avoid, and acceptance ratio = 1.5; the data of the first 40 iterations of the experiment are shown in Figure 17.

Figure 17 shows that in the cultural particle swarm curve, according to the situational knowledge and normative knowledge, the rate of particles toward the global optimal solution is gradually increased with the increase of the number of population iterations. At the 9-th iteration, the global optimal particle position is found, and the fitness value is 0.268394533746147 (see Figure 18). When the particle iterates to the 8-th time, the particle position exceeds the global optimal solution, but according to the knowledge and particle swarm velocity update formula, the particle quickly returns to the global optimal solution position again, which indicates that this algorithm has good global search capabilities.

In order to analyze the global search ability and convergence performance of the cultural particle swarm algorithm in this section, a comparison experiment is performed between the algorithm in this paper and the standard PSO, that is, the PSO algorithm, GA (genetic) algorithm, and AFSA (fish school) algorithm. The results are shown in Figure 19.

Figure 19 shows the display of the four algorithms for finding the optimal steering angle of the ship. It can be seen from the figure that after 40 iterations, the CPSO (cultural particle swarm optimization) algorithm adopted in this paper is better than the standard PSO, GA, and AFSA algorithms. This is because the belief space of the cultural particle swarm algorithm uses the influence function to guide the population space, enhances the diversity of particles in the population space, and uses a dynamic inertia weight adjustment strategy to adaptively adjust weight according to the degree of population convergence, making the algorithm have a strong global optimizing ability.

The standard PSO algorithm reached the optimal solution the 24-th iteration, but because the linearly decreasing inertial weight is used, the inertial weight change is too single, and the process of finding the optimal steering angle of the ship is nonlinear and does not consider the population iteration information. The adjustment ability of the standard PSO algorithm is limited, so the algorithm has decreased compared with the cultural particle swarm algorithm. The GA and AFSA algorithms reached the optimal solution in the 26-th and 16-th generations, respectively, but because there is no influence function to guide the population space and it has a certain degree of blindness to the population iteration, the convergence speed has decreased compared to the cultural particle swarm algorithm.

(1) Initialize particle population size N, dimension 1, iterations item, steering angle search space $[-x_{\max}, x_{\max}]$, particle maximum velocity $v_{\max}$.
(2) Initialize the population space, particle initial position x, initial velocity v, initial $<S, N>$.
(3) Calculate the objective value of population space particles, particle' current optimal position $p\text{best}_i$, global optimal position $g\text{best}$.
(4) Update inertia weights according to formula (21), and equation (19) updates particle velocity and position, by x_i update $p\text{best}_i$, $g\text{best}$.
(5) Replace the least fit particles in the belief space with the best fit particles in the population space based on acceptance function and update the position of the population space particles by the influence function.
(6) Determine whether the maximum number of iterations item has been reached; if not, go to step 4; otherwise, go to step 7.
(7) Output $g\text{best}$, algorithm ends.

Algorithm 3: Optimal steering angle decision algorithm based on cultural particle swarm.

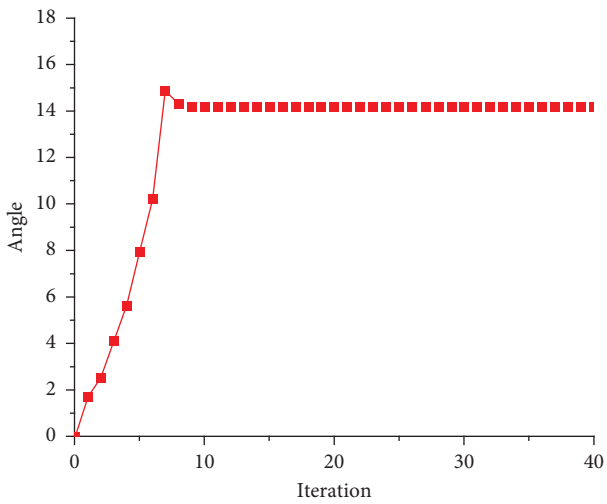


Figure 17: Cultural particle swarm curve.

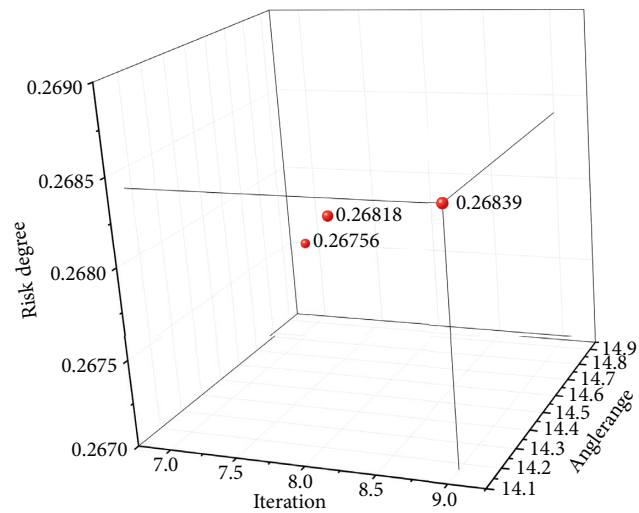


Figure 18: Algorithm's three-dimensional figure (7–9 generations).

6. Electronic Chart Platform Integration and Experimental Analysis

We integrate the method proposed in this paper into the electronic chart platform. The electronic chart platform is used as the verification environment of the method. At the same time, the AIS data of the ship are displayed on the electronic chart platform and provided to the related algorithm. Electronic chart platform is an important navigation tool chart for ships and other marine vehicles, which can provide true and complete environmental information required during navigation, mainly including land, ocean, water depth, obstacles, and islands. So, it is widely used in ocean path planning and other aspects.

In this paper, the international standard electronic chart platform developed in C ++ language is used as the experimental verification environment, as shown in Figure 20. The system has the following characteristics:

(1) The system uses international standards to display chart information, and we can zoom and observe the chart interface at any scale.

(2) The system uses the MFC (Microsoft Foundation Class) framework to generate a dynamic link library of related algorithms, providing a flexible and convenient interface design.

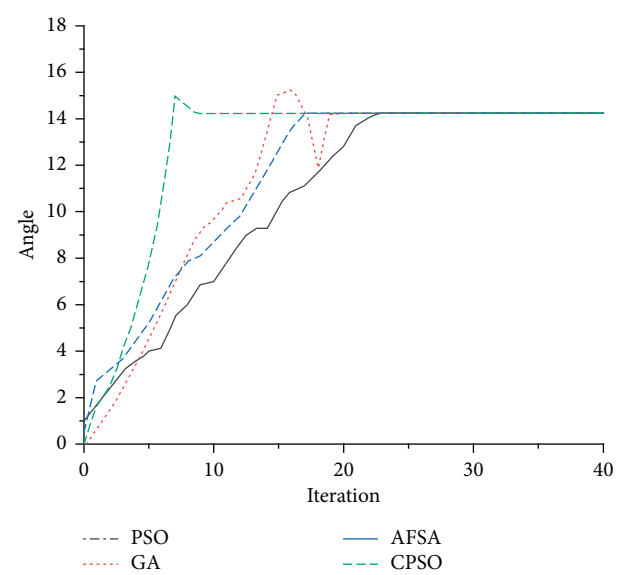


Figure 19: Comparative experiment chart.

Figure 20: Electronic chart simulation environment.

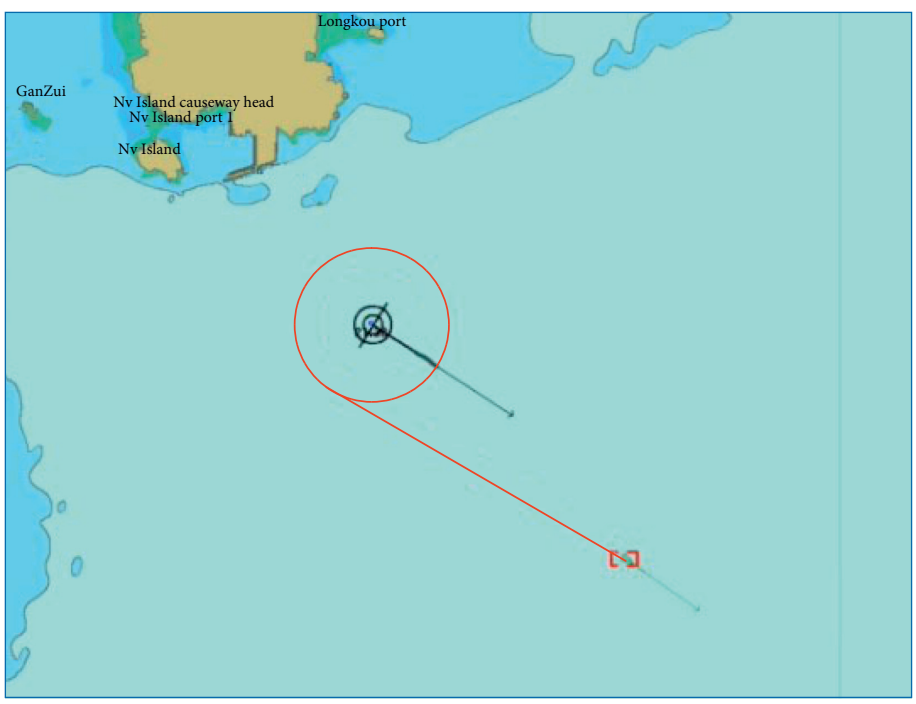


Figure 21: Own ship is overtaking ship.

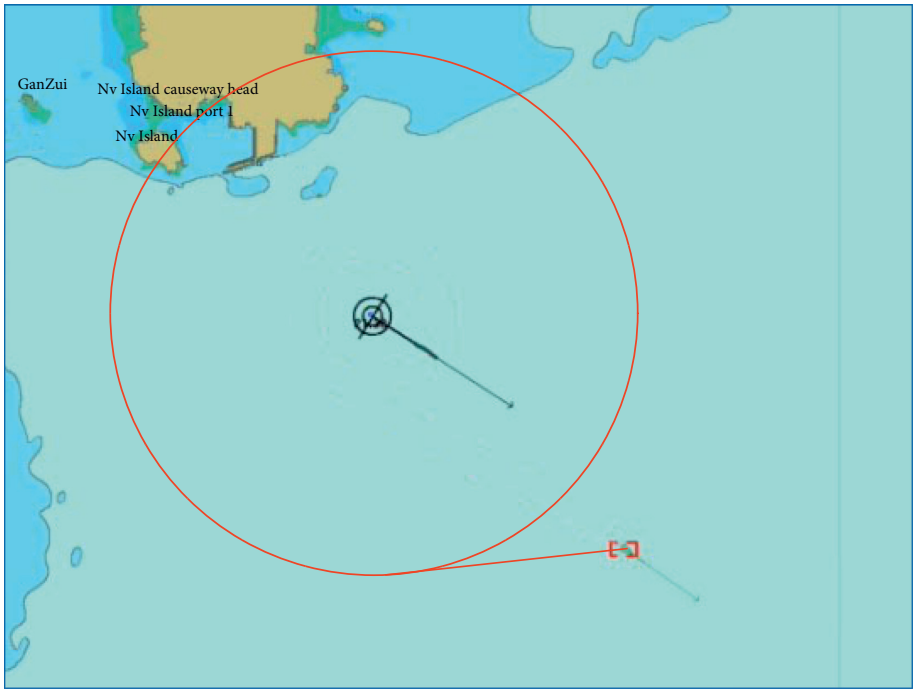


Figure 22: Right-steering collision avoidance.

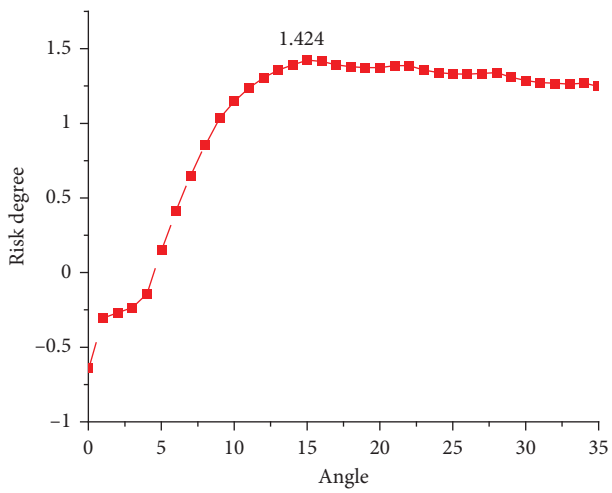


Figure 23: The hazard curve of ship collision avoidance described in this scene.

(3) The system provides operations for autonomously setting the movement information of the current ship and other ships, including information such as longitude, latitude, heading, and speed, and all ship models in the system use bulk carriers.

(4) It can dynamically display the movement tracks of all ships in the chart, which is convenient to observe the simulation results of real environment. The interface is shown in Figure 20.

6.1. Steering Angle Direction Decision Verification. Figures 21–29 show the ship encounter scenarios, verifying whether the model meets the rules of collision avoidance. Figures 21 and 22 are the scenes of own ship as overtaking ship. The own ship speed is 15 knots and the target ship speed is 8 knots. Besides that, the specific parameters of the ship are shown in Table 3.

There is a danger of collision in the current scene. The analysis in Section 4.3 shows that the steering should be driven to the left. Figure 23 shows the collision risk curve of the ship depicted in the scene. As the ship steers to the left (the left direction is counterclockwise and the sign is negative), the ship's collision risk curve gradually increases first, reaching the highest point of the curve at 1.424 at 15.2 degrees, and then it almost stabilized in the 1.42 to 1.20 range. Therefore, the ship steers 15.2 degrees to the left to

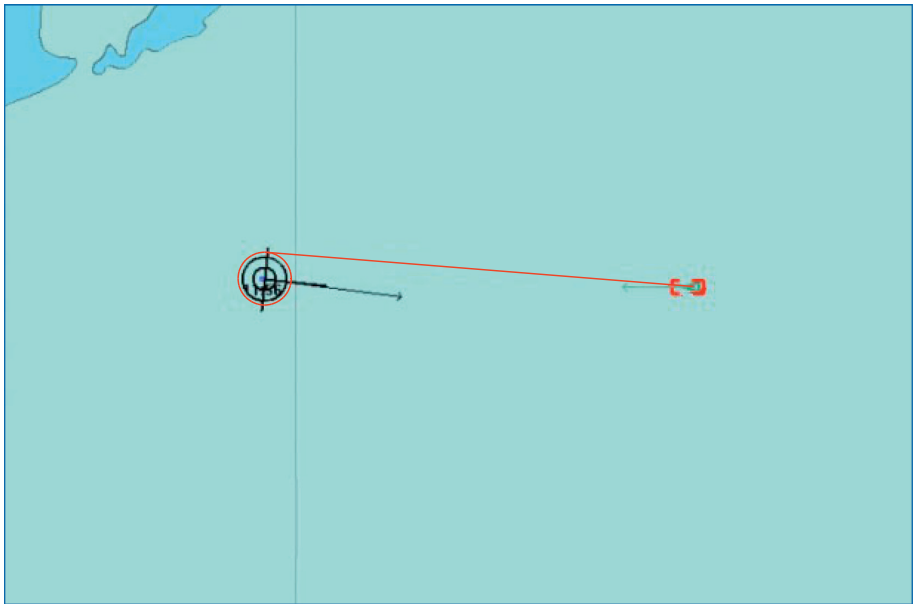


Figure 24: Own ship is facing the ship.

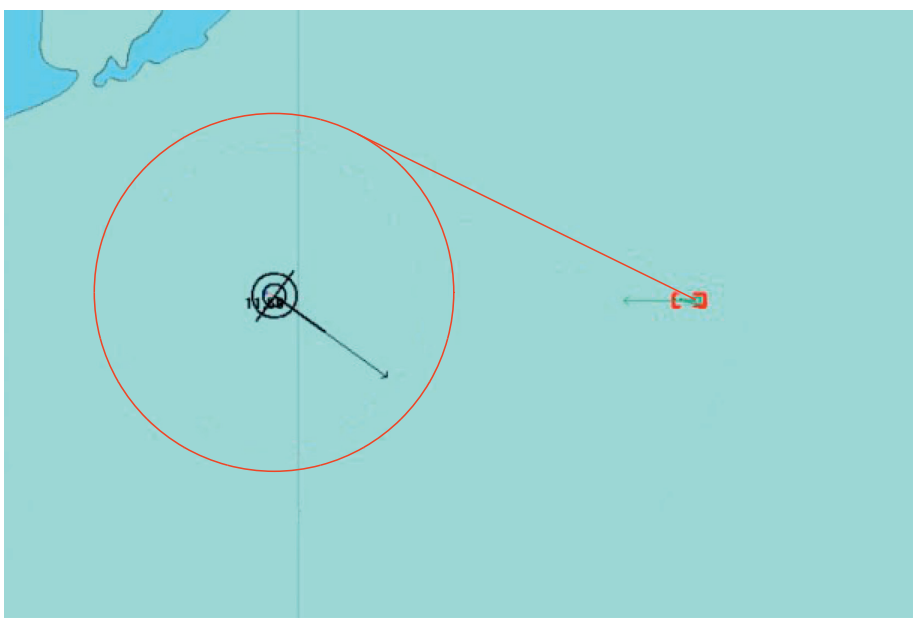


Figure 25: Right-steering collision avoidance.

avoid collisions, and the experimental results meet the requirements of the ship collision avoidance rules.

Figures 24 and 25 are the encounter scenes of the ship. The speed of the ship is 15 knots and the speed of the target ship is 8 knots. The specific parameters of the ship are shown in Table 4.

There is a danger of collision in the current scene. The analysis in Section 4.3 above shows that the steering should be driven to the right. Figure 26 shows the collision risk curve of the ship depicted in the scene. As the ship steers to the right (the clockwise direction is positive), the ship's collision risk curve gradually increases first, reaching the highest point of the curve at 27.8 degrees at 1.126, and then slightly drops to around 1.0. Therefore, the ship steers 27.8 degrees to the right to avoid the collision, and the experimental results meet the requirements of the ship collision avoidance rules.

Figures 27 and 28 are the crossing situation of the ship, and the specific parameters of the ship are shown in Table 5.

There is a danger of collision in the current scene. The analysis in 4.3 above shows that the rudder should be driven to the right. The collision risk curve of own ship depicted in the scene is shown in Figure 29. As the ship steers to the right, the collision risk curve of the ship gradually increases,

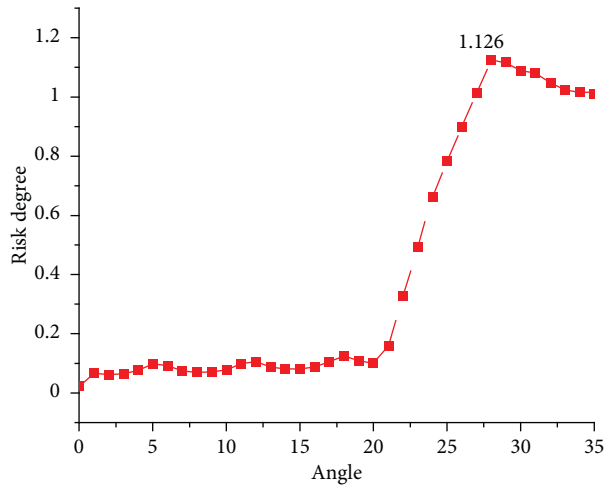


Figure 26: The hazard curve of ship collision avoidance described in this scene.

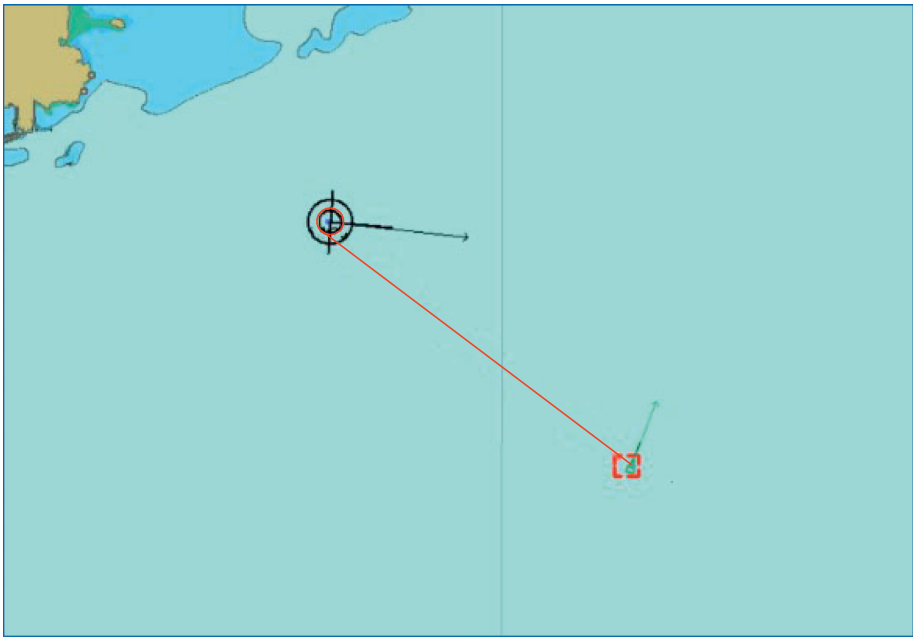


Figure 27: Own ship is a give way ship.

reaching the highest point 1.066 of the curve at 35 degrees. Therefore, own ship steers 35 degrees to the right to avoid the collision, and the experimental results meet the requirements of the ship collision avoidance rules.

6.2. Steering Angle Decision Verification. We use the ship steering collision avoidance model proposed in this paper to carry out the collision avoidance simulation experiment. The simulation environment diagram is shown in Figure 30. In the simulation initial conditions, the initial course of the ship is 92.0 degrees, the speed is 15 knots, the longitude is 120.87 degrees, and the latitude is 36.35 degrees; Ship-1 is the first target ship, its course is 270.0 degrees, its speed is 7 knots/hour, longitude is 120.87 degrees, and the latitude is 36.35 degrees; Ship-2 is the second target ship, its course is 40.0 degrees, the speed is 8 knots/hour, the longitude is 120.90 degrees, and the latitude is 36.29 degrees; Ship-3 is the third target ship, its course is 190.0 degrees, its speed is 9 knots, longitude is 120.87 degrees, and its latitude is 36.35 degrees.

Figures 31(a) and 31(b) are the situations where the ship and the three target ships will encounter the situation. By analyzing the system log of the collision avoidance module in the scenario in Figure 31, the collision risk change during the collision avoidance process is obtained. The drawn curve is shown in Figure 32.

In the initial scenario, own ship and Ship-1 are facing each other, and there is a danger of collision. The analysis in 4.3 above shows that own ship should steer to the right at this

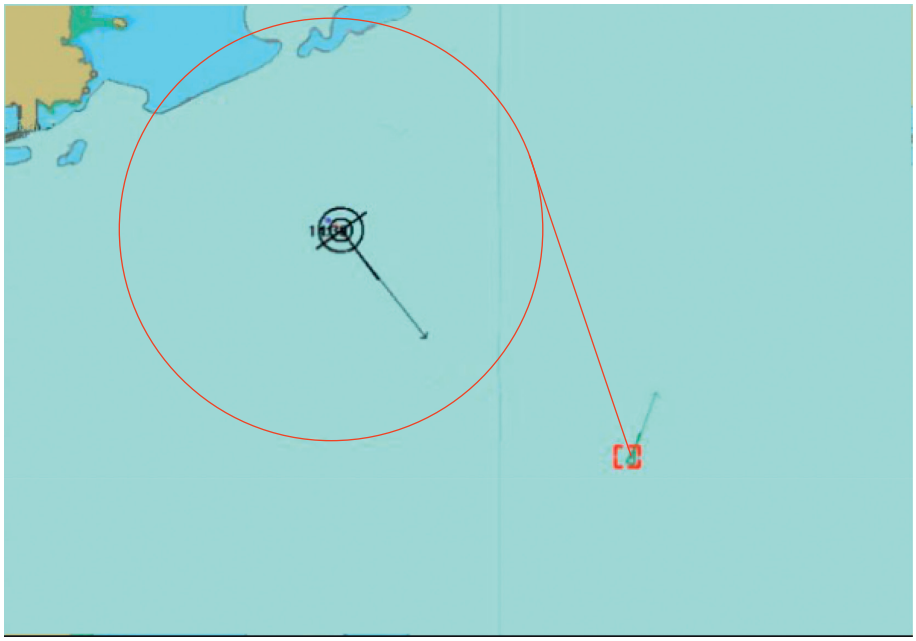


Figure 28: Right-steering collision avoidance.

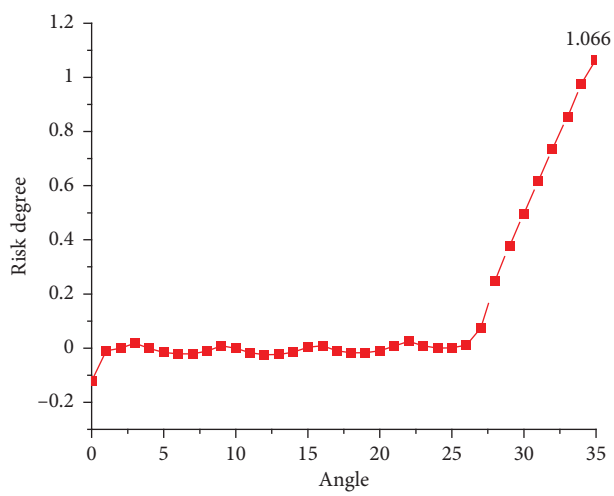


Figure 29: The hazard curve of ship collision avoidance described in this scene.

Table 3: Relevant data of chase scene experiment.

Related parameters	Value
Position	132.85°
Relative position	10.75°
Relative velocity	7.02 n mile/h
Relative speed heading	298.80°
DCPA	−0.77/n mile
TCPA	26.32/min
Relative heading	2.90°
Ship spacing	3.48 n mile
Current risk	−0.62

time. Figure 32 shows the ship collision risk curve depicted in the scene. As the ship steers to the right, the collision hazard curve of own ship and Ship-1 gradually increases, while the hazard curve with Ship-2 gradually decreases. This is because own ship steers to the right, causing the ship to move closer to Ship-2, which increases the risk of collision.

Table 4: Relevant data of encounter scene experiment.

Related parameters	Value
Position	91.0°
Relative position	0.70°
Relative velocity	23.0 n mile/h
Relative speed heading	270.20°
DCPA	−0.07/n mile
TCPA	12.51/min
Ship spacing	2.83 n mile
Current risk	0.02
Relative heading	179.70°

Table 5: Cross-encounter scene experimental data.

Related parameters	Value
Position	129.28°
Relative position	38.48°
Relative velocity	14.50 n mile/h
Relative speed heading	302.21°
DCPA	−0.53/n mile
TCPA	17.76/min
Relative heading	70.80°
Ship spacing	3.23 n mile
Current risk	−1.18

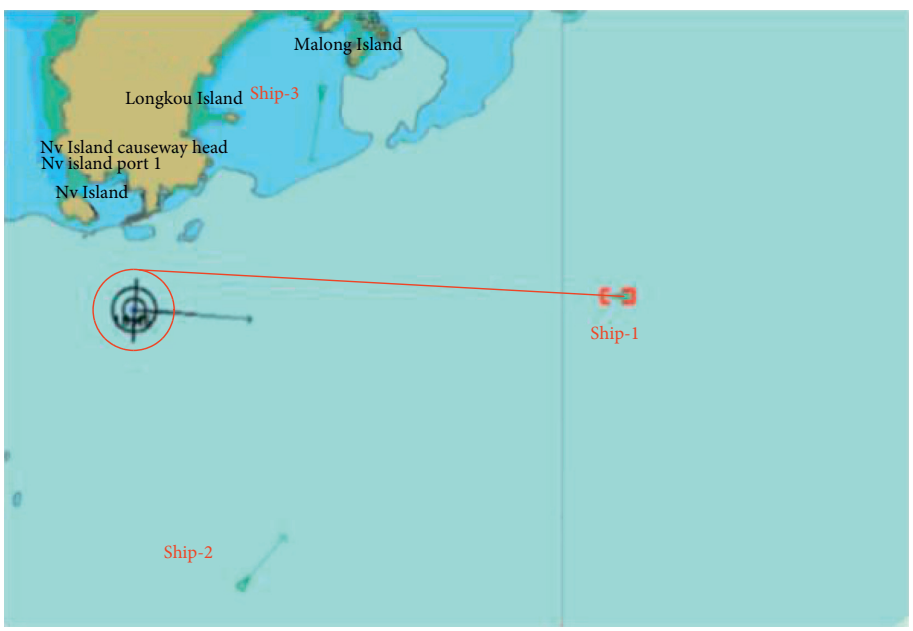
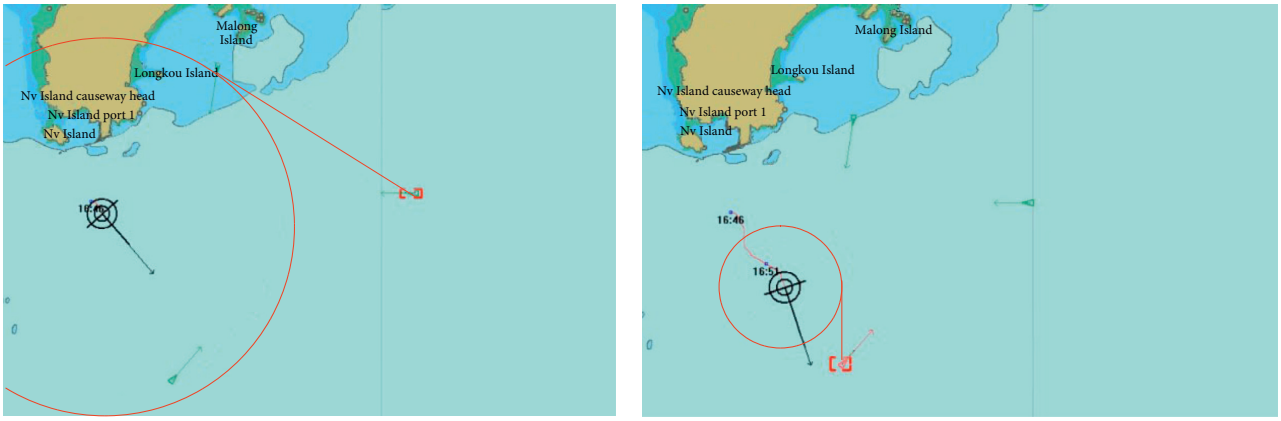


Figure 30: Initial simulation environment diagram.

The hazard curve of the own ship and Ship-3 is relatively stable above 1.4, and there is no danger of collision. The hazard curve of Ship-2 and Ship-1 intersects at 22.5 degrees, so own ship steers 22.5 degrees to the right to avoid the collision. The collision avoidance state is shown in Figure 31(a). After evading Ship-1 (the most dangerous ship at present), evade Ship-2. The collision avoidance state is shown in Figure 31(b).

As shown in Figure 33(a), the own ship is ready to take action to resume sailing after avoiding the dangerous target ship and keeping clear. As shown in Figure 33(b), the position of own ship finally returned to the original route. Hence, the results indicate that the own ship can take proper action to avoid other target ships and has a better performance. During the entire experiment, the ship can avoid other target ships reasonably and has a better avoidance effect.



(a) (b)

Figure 31: The encounter situation between the ship and the three target ships. (a) Avoid target Ship-1. (b). Avoid target Ship-2.

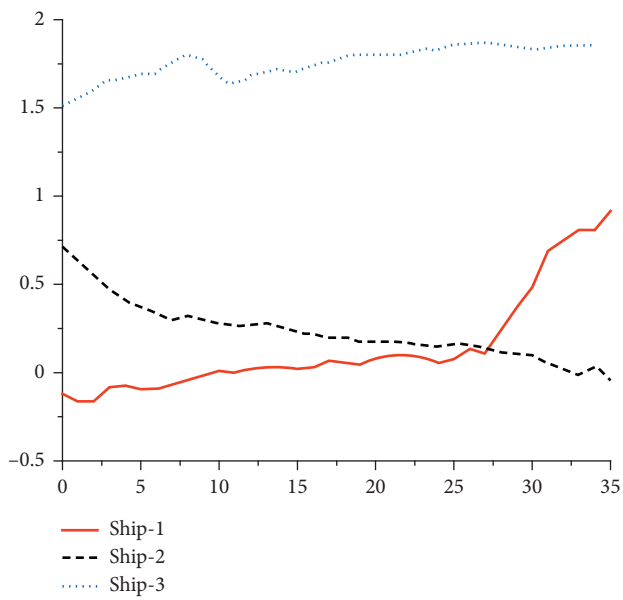


Figure 32: Change curve of collision risk during collision avoidance.

(a) (b)

Figure 33: Picture of the end of own ship avoidance. (a) Location map of own ship after evading Ship-2. (b) End of own ship avoidance.

7. Conclusions

In this paper, a decision-making method for ship collision avoidance based on improved cultural particle swarm is proposed. Aiming at the fact that traditional evolutionary algorithms do not use the information of population iteration which leads to the slow adjustment ability and convergence speed of the algorithms, the cultural particle swarm is improved and applied to collision avoidance system in this paper. Firstly, AIS data of ship are acquired based on electronic chart platform. However, in the actual AIS ship data transmission, the update is not timely and the error is large. Hence, the Kalman filter is used to smooth and predict the ship trajectory in this paper. Secondly, based on the fuzzy distribution method, the danger degree of the ship is estimated and the steering angle direction is determined by combining the COLREGs and the encounter status of the ship. Finally, the optimal steering angle of the ship is found by cultural particle swarm optimization algorithm to realize the decision making of ship collision avoidance.

The complex situations of three ships are simulated in electronic chart. The results show that the ship adopts the optimal and reasonable steering angle decision in this encounter scene and realizes the autonomous collision avoidance. The cultural particle swarm optimization algorithm adopted in this paper is compared with other three evolutionary methods. Experiments demonstrate that the cultural particle swarm algorithm has faster convergence speed and higher accuracy and can achieve continuous job output, which further validates the effectiveness of the algorithm. However, the speed decision of ships is not considered in this paper, and there are a large number of collision avoidance scenarios for ships at sea. The test scenarios in this paper are not comprehensive enough. How to consider the decision of ship steering angle and ship speed, as well as to divide and test the scene in a complex encounter situation, is the focus of the next step of this thesis.

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.

Acknowledgments

This study was supported by the National Key R&D Program of China (grant no. 2018YFB1601502), the National Natural Science Foundation of China (grant no. 51679025), the Liaoning Revitalization Talents Program (grant no. XLYC1902071), and the Fundamental Research Funds for the Central Universities (grant nos. 3132019313 and 3132019354).

References

[1] A. Lazarowska, "Decision support system for collision avoidance at sea," *Polish Maritime Research*, vol. 19, pp. 19–24, 2012.

[2] S. Li, J. Liu, and R. R. Negenborn, "Distributed coordination for collision avoidance of multiple ships considering ship maneuverability," *Ocean Engineering*, vol. 181, pp. 212–226, 2019.

[3] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. H. A. J. M. van Gelder, "Ship collision avoidance methods: state-of-the-art," *Safety Science*, vol. 121, pp. 451–473, 2020.

[4] T. Statheros, G. Howells, and K. M. Maier, "Autonomous ship collision avoidance navigation concepts, technologies and techniques," *Journal of Navigation*, vol. 61, no. 1, pp. 129–142, 2008.

[5] W. Shaobo, Z. Yingjun, and L. Lianbo, "A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method," *Ocean Engineering*, vol. 215, p. 107910, 2020.

[6] C. Tam, R. Bucknall, and A. Greig, "Review of collision avoidance and path planning methods for ships in close range encounters," *Journal of Navigation*, vol. 62, no. 3, pp. 455–476, 2009.

[7] X. Wang, Z. Liu, and Y. Cai, "The ship maneuverability based collision avoidance dynamic support system in close-quarters situation," *Ocean Engineering*, vol. 146, pp. 486–497, 2017.

[8] H. Chen, M. Wang, and X. Zhao, "A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems," *Applied Mathematics and Computation*, vol. 369, p. 124872, 2020.

[9] S. Wu, *Research on Improvement and Application of Multi-Objective Particle Swarm Optimization Algorithm*, Jiangnan University, Wuxi, China, 2013.

[10] G. Yuan, Z. Wei, and Y. Yang, "The global convergence of the Polak-Ribière-Polyak conjugate gradient algorithm under inexact line search for nonconvex functions," *Journal of Computational and Applied Mathematics*, vol. 362, pp. 262–275, 2019.

[11] J. Guo, W. Yuan, X. Dang, and M. S. Alam, "Cable force optimization of a curved cable-stayed bridge with combined simulated annealing method and cubic B-Spline interpolation curves," *Engineering Structures*, vol. 201, Article ID 109813, 2019.

[12] A. M. Mohammed and S. O. Duffuaa, "A tabu search based algorithm for the optimal design of multi-objective multi-product supply chain networks," *Expert Systems with Applications*, vol. 140, Article ID 112808, 2020.

[13] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, vol. 23, no. 9, pp. 3137–3166, 2019.

[14] Y. Lu, C. A. Phillips, and M. A. Langston, "A robustness metric for biological data clustering algorithms," *BMC Bioinformatics*, vol. 20, no. 15, pp. 1–8, 2019.

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

[16] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 4104–4108, IEEE, Orlando, FL, USA, October 1997.

[17] G.-C. Luh, C.-Y. Lin, and Y.-S. Lin, "A binary particle swarm optimization for continuum structural topology optimization," *Applied Soft Computing*, vol. 11, no. 2, pp. 2833–2844, 2011.

[18] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, pp. 1945–1950, IEEE, Washington, DC, USA, July 1999.

[19] J. Riget and J. S. Vesterstrøm, "A diversity-guided particle swarm optimizer-the ARPSO," Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep, 2002.

[20] M. Daneshyari and G. G. Yen, "Cultural-based multiobjective particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, p. 553, 2011.

[21] X.M. Tao and L. Yang, "Cultural particle swam optimization algorithm with adaptive guidance," *Computer Engineering and Application*, vol. 47, no. 14, pp. 37–41.

[22] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, 2016.

[23] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with COLREGS, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2013.

[24] D. Kim, K. Hirayama, and T. Okimoto, "Distributed stochastic search algorithm for multi-ship encounter situations," *Journal of Navigation*, vol. 70, no. 4, pp. 699–718, 2017.

[25] A. Lazarowska, "Ant colony optimization based navigational decision support system," *Procedia Computer Science*, vol. 35, pp. 1013–1022, 2014.

[26] H. D. Nguyen, N. V. Do, V. T. Pham, A. Selamat, and E. Herrera-Viedma, "A method for knowledge representation to design intelligent problems solver in mathematics based on rela-ops model," *IEEE Access*, vol. 8, pp. 76991–77012, 2020.

[27] C. Tam and R. Bucknall, "Path-planning algorithm for ships in close-range encounters," *Journal of Marine Science and Technology*, vol. 15, no. 4, pp. 395–407, 2010.

[28] D. X. Liu, J. P. Zhang, and F. W. Wang, "Decision model of encountering situations in ship intelligent collision avoidance decision system," *Marine Technology*, no. s1, pp. 118–123, 2004.

[29] A. Goudossis and S. K. Katsikas, "Towards a secure automatic identification system (AIS)," *Journal of Marine Science and Technology*, vol. 24, no. 2, pp. 410–423, 2019.

[30] L. Yan, D. P. Roy, Z. Li, H. K. Zhang, and H. Huang, "Sentinel-2A multi-temporal misregistration characterization and an orbit-based sub-pixel registration methodology," *Remote Sensing of Environment*, vol. 215, pp. 495–506, 2018.

[31] X. Nie, "Detection of grid voltage fundamental and harmonic components using Kalman filter based on dynamic tracking model," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1191–1200, 2019.

[32] Y. Hu, A. Zhang, W. Tian, J. Zhang, and Z. Hou, "Multi-ship collision avoidance decision-making based on collision risk index," *Journal of Marine Science and Engineering*, vol. 8, no. 9, p. 640, 2020.

[33] L. Chen, *Research on Intelligent Collision Avoidance Assisted Decision-Making for Maritime Multi-Target Ships*, Wuhan University of Technology, Wuhan, China, 2011.

[34] L. Li, Z. Xiong, Y. Gao, and Q. Ren, "Generation and optimization of intelligent decision-making for collision avoidance of single ship," *China Navigation*, vol. 2, pp. 49–52, 2002.

[35] J. Zhou and C. Wu, *Construction of Ship Collision Risk Model*, vol. 2004, no. 1, pp. 61–65, 2004.

[36] M. Jafari, E. Salajegheh, and J. Salajegheh, "An efficient hybrid of elephant herding optimization and cultural algorithm for optimal design of trusses," *Engineering with Computers*, vol. 35, no. 3, pp. 781–801, 2019.

[37] G. Beni, "The concept of cellular robotic system," in *Proceedings IEEE International Symposium on Intelligent Control 1988*, pp. 57–62, IEEE, Arlington, VA, USA, August 1988.

[38] N. Yang and Y. Jing, "Research on function extreme value optimization based on improved PSO algorithm," *Computer Simulation*, vol. 32, no. 9, pp. 263–266, 2015.

[39] M. Z. Ali, N. H. Awad, R. G. Reynolds, and P. N. Suganthan, "A balanced fuzzy cultural algorithm with a modified levy flight search for real parameter optimization," *Information Sciences*, vol. 447, pp. 12–35, 2018.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8882554, 14 pages
https://doi.org/10.1155/2020/8882554

Research Article

A Real-Time Train Timetable Rescheduling Method Based on Deep Learning for Metro Systems Energy Optimization under Random Disturbances

Jinlin Liao,[1] **Feng Zhang**,[1] **Shiwen Zhang**,[1] **and Cheng Gong**[2]

[1]*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2]*School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30318, USA*

Correspondence should be addressed to Feng Zhang; fzhang@sjtu.edu.cn

Received 25 August 2020; Revised 21 November 2020; Accepted 30 November 2020; Published 12 December 2020

Academic Editor: Petr Dolezel

Considering that uncertain dwell disturbances often occur at metro stations, researchers have proposed many methods for solving the train timetable rescheduling (TTR) problem. This paper proposes a Modified Genetic Algorithm-Gate Recurrent Unit (MGA-GRU) method, which is a real-time TTR method based on deep learning. The proposed method takes the Gate Recurrent Unit (GRU) network as the decision network and uses the results produced by the Modified Genetic Algorithm (MGA) as the training set of the decision network. A well-trained decision network can provide effective solutions in real time after random disturbances occur, in order to optimize the net traction energy consumption of trains in metro systems. Based on the Shanghai Metro Line One (SML1) pilot network, this paper establishes a comprehensive model of the metro system as a training and testing environment to verify the energy-saving effect and real-time performance of the proposed method in solving the TTR problem. The experimental results show that in the two-train metro system, the three-train metro system, and the five-train metro system, the MGA-GRU method can save an average of energy by 4.45%, 6.16%, and 7.19%, while the average decision time is only 0.15 s, 0.27 s, and 0.33 s, respectively.

1. Introduction

Compared with ground transportation such as buses and taxis, urban metro systems have achieved rapid development worldwide due to the advantages of no traffic jams, large capacity, and high safety [1]. Although metro systems are energy-efficient compared with other ground vehicles, they still consume a lot of energy [2, 3]. Due to problems such as rising energy prices and environmental pollution, in recent years, reducing the net traction energy consumption of trains in metro systems by studying and optimizing the train timetable has become an important research topic [4, 5].

In metro systems, when a train brakes, it can regenerate braking energy to the system [6]. Regenerative braking energy (RBE) can be reused by trains that are simultaneously accelerating and can also be stored in energy storage devices such as batteries [7], supercapacitors [8], and flywheels [9]. Otherwise, RBE must be consumed by resistors as thermal energy to prevent the train voltage from surpassing the safety threshold [10, 11]. The timetable of a metro system can be either predetermined offline or dynamically changed in real time. By designing a suitable timetable, the acceleration trains and the braking trains can be better synchronized to make better use of RBE [12].

Although researchers have conducted extensive research on train timetable in the past few decades, trains in metro systems are still often subject to unexpected disturbances, such as a sudden increase in passenger flow, unexpected accidents, and unplanned parking [13, 14]. To solve this kind of problem, researchers have proposed many train timetable rescheduling (TTR) methods [15–19].

Previous studies on the TTR problem have focused on reducing the delay time caused by disturbances, which can be divided into two categories: one is to minimize the total

delay time of passengers [20]; the other is to minimize the total delay time of all trains [21]. Šemrov et al. [22] introduced a real-time TTR method based on Q-learning. The proposed method has carried out a large number of experiments on the real-world railway network in Slovenia. The experimental results show that the solutions of Q-learning are at least equivalent and generally superior to simple first-in-first-out (FIFO) and random walk methods that do not rely on learning agents.

Different from these two types of traditional methods, the TTR method studied in this paper aims to reschedule train timetable after disturbances occur to reduce traction energy consumption. Hou et al. [23] developed a mixed-integer programming (MIP) model to solve a metro train timetable rescheduling problem, which aims to jointly optimize the total train delay, the number of stranded passengers, and the energy consumption of trains. Zhao et al. [24] implemented three search methods, namely, enhanced brute force (EBF), ant colony optimization (ACO), and Genetic Algorithm (GA), to minimize energy consumption and delay after being disturbed. The results show that these three methods can find close to the best or the best train trajectories and driving styles to reduce the energy or improve safety and passenger's comfort. Gong et al. [25] proposed a Compensational Driving Strategy Algorithm (CDSA) to restore the disturbed train to the original optimal timetable by reducing the travel time of the disturbed train in the next section after a disturbance occurs. The results show that compared with not using CDSA after a disturbance occurs, using CDSA can save 1.86% of energy on average.

However, these optimization methods (EBF, ACO, and GA) implemented by Zhao et al. are not suitable for solving the TTR problem in real time due to the long calculation time. And the CDSA proposed by Gong et al. only rearranges the coasting speed of the disturbed trains, which does not adjust other trains' coasting speeds and all trains' dwell time. In response to these problems, this paper proposes a TTR method based on deep learning, called Modified Genetic Algorithm-Gate Recurrent Unit (MGA-GRU) by combining the modified Genetic Algorithm (MGA) with the Gate Recurrent Unit (GRU) network.

Up to now, many methods based on a general GA have been proposed to solve scheduling and optimization problems [26–29]. Corresponding experimental results show that these methods can find high-quality solutions for large-scale case. And GRU has been applied to solve problems with time-series dimensions [30, 31]. These experimental results show that GRU can extract more rich and complex information from sequences and aspects.

Better than EBF, ACO, and GA, MGA-GRU can reschedule the timetable in real time after random disturbances occur. Unlike CDSA which only rearranges the coasting speed of the disturbed train, MGA-GRU rearranges the coasting speed and dwell time of all trains in the metro network in real time after disturbances occur, so as to achieve better energy-saving effect.

The remainder of this paper is organized as follows. Section 2 builds three models based on Shanghai Metro Line One (SML1). Section 3 introduces the MGA-GRU method to solve the TTR problem in real time after a disturbance occurs. In Section 4, four experiments based on the SML1 pilot network are conducted to verify the energy-saving effect and real-time performance of the proposed method. Section 5 concludes this paper.

2. Modeling

In this section, three models are proposed to formulate the metro system: time model, mechanical model, and power model.

For a better understanding of this paper, the assumptions, decision variables, and parameters are first introduced.

2.1. Assumptions

(1) The distance between two adjacent stations of SML1 is relatively small. According to the actual operation of trains on the SML1 and the description of Su et al. [32], each train adopts a single-cycle acceleration-coasting-braking strategy instead of repeated acceleration and braking.

(2) Dwell disturbances are small enough so as not to lead to network disruption.

(3) From the first train's departure to the last train's arrival, only one dwell disturbance occurs. But the value of the disturbance is random.

2.2. Decision Variables

$v_c^{m,n}$: coasting speed of train no. m from station no. $n-1$ to station no. n

$t_{\mathrm{dw}}^{m,n}$: dwell time of train no. m at station no. n

2.3. Parameters

m : index of train

n : index of station

M : total number of trains

N : total number of stations

$t_{\mathrm{tr}}^{m,i}$: travel time of train no. m from station no. $i-1$ to station no. i

$t_{\mathrm{de}}^{m,n}$: departure instant of train no. m at station no. n

$t_{\mathrm{dw}}^{m,n}$: dwell time of train no. m at station no. n

t_1 : Headway

ε : Duration of a dwell disturbance

$t_{\mathrm{total}}^{M,N}$: total time

v : speed of the train

F_T : traction force

F_B : braking force

F_R : running resistance

F_G : gravity

θ : track slope

x^m : position of train no. m

$P_T^{m,n}$: traction power of train no. m from station no. $n-1$ to station no. n

$P_R^{m,n}$: regenerative braking power of train no. m from station no. $n-1$ to station no. n

$P_F^{m,n}$: feedback power of train no. m from station no. $n-1$ to station no. n

η_1 : conversion efficiency of the train traction system (from electrical energy to mechanical energy)

η_2 : conversion efficiency of the train braking system (from mechanical energy to electrical energy)

η_3 : braking energy feedback coefficient

E_T : traction energy consumption of acceleration trains

E_F : regenerative braking feedback energy used by acceleration trains

E : net energy consumption

2.4. Time Model. The time model defines the departure instant t_{de}, travel time t_{tr}, and dwell time t_{dw} of each train at each station [33]. The starting station is defined as station no. 1. The instant when train no. 1 leaves the starting station is defined as time = 0. The interval between each adjacent train leaving the starting station is equal.

If a disturbance occurs at station no. n of train no. m, the corresponding dwell time will increase from $t_{\mathrm{dw}}^{m,n}$ to $t_{\mathrm{dw}}^{m,n}+\varepsilon$. Then the departure instant of train no. m at station no. n is

$$t_{\mathrm{de}}^{m,n} = t_{\mathrm{de}}^{m,1} + \sum_{i=2}^{n}\left(t_{\mathrm{tr}}^{m,i} + t_{\mathrm{dw}}^{m,i}\right) + \varepsilon. \tag{1}$$

According to the assumptions above, only one disturbance occurs during each entire test procedure. Therefore, the instant $t_{\mathrm{total}}^{M,N}$ when the last train arrives at the terminal is defined as

$$t_{\mathrm{total}}^{M,N} = t_{\mathrm{de}}^{M,1} + \sum_{i=2}^{N-1}\left(t_{\mathrm{tr}}^{M,i} + t_{\mathrm{dw}}^{M,i}\right) + t_{\mathrm{tr}}^{M,N} + \varepsilon. \tag{2}$$

2.5. Mechanics Model. According to the assumption above, each train adopts a single-cycle acceleration-coasting-braking strategy instead of repeatedly acceleration and braking [25]. The unit of F_T, F_R, F_B, etc., is N.

In the acceleration phase, $F_B = 0$; the relationship between F_T and speed v is shown in the following equation:

$$F_T = \begin{cases} 550, & 0 < v \le 10\,\mathrm{m/s}, \\ \dfrac{19800}{v}, & v > 10\,\mathrm{m/s}. \end{cases} \tag{3}$$

When the speed is lower than 10 m/s, the train is in a constant torque traction state, and the acceleration of the train is a fixed value. When the speed increases beyond 10 m/s, the train switches to a constant power traction state. In this state, the traction power is a fixed value, and the traction force is inversely proportional to the speed.

In the coasting phase, $F_B = 0$, $F_T = 0$; the relationship between F_R and v conforms to the Davis equation [34]:

$$F_R = 7.398 + 0.255v + 0.012v^2. \tag{4}$$

In the braking phase, $F_T = 0$; the relationship between F_B and v is shown in the following equation:

$$F_B = \begin{cases} 550, & 0 < v \le 18.056\,\mathrm{m/s}, \\ \dfrac{35750}{v}, & v > 18.056\,\mathrm{m/s}. \end{cases} \tag{5}$$

When the speed is higher than 18.056 m/s, the braking force is inversely proportional to the speed. When the speed decreases within 18.056 m/s, the deceleration is a fixed value.

2.6. Power Model. In the metro system, there are three driving states of trains: acceleration, coasting, and braking. Accelerating trains convert electrical energy into mechanical energy, while braking trains can regenerate mechanical energy into electrical energy. The electric energy generated by the braking trains can be supplied to the acceleration trains. This implies that if the trains can be arranged with an appropriate strategy, a lot of energy can be saved by the use of this part of RBE.

P_B, the regenerative braking power, is defined as

$$P_B = \eta_1 F_B v. \tag{6}$$

P_T is the traction power and is defined as

$$P_T = \frac{F_T v}{\eta_2}. \tag{7}$$

If the braking power is less than the traction power, it can be fully used; otherwise, resistors will kick in and consume the overflowing braking power to maintain the train voltage under a safe value. The minimum value of traction power and braking conversion power is defined as P_F [33]:

$$P_F = \min\left(P_T, \eta_3 P_B\right) = \min\left(\frac{F_T v}{\eta_2}, \eta_3\eta_1 F_B v\right). \tag{8}$$

2.7. Relationship between Coasting Speed and Travel Time. The area enclosed by the speed curve and the time axis is the distance between two adjacent metro stations. As shown in Figure 1, if the acceleration and driving strategy is determined, the coasting speed and the travel time form a one-to-one mapping between two adjacent stations. Higher coasting speed corresponds to a shorter travel time. Therefore, the travel time can be controlled by controlling the coasting speed. And the relation can be defined as $t_{\mathrm{tr}}^{m,n} = f\left(v_c^{m,n}\right)$ [25].

3. Energy Optimization under Disturbances

In order to optimize the net traction energy consumption in real time after a dwell disturbance occurs, this paper proposes an MGA-GRU method based on deep learning. This method combines the modified Genetic Algorithm (MGA)

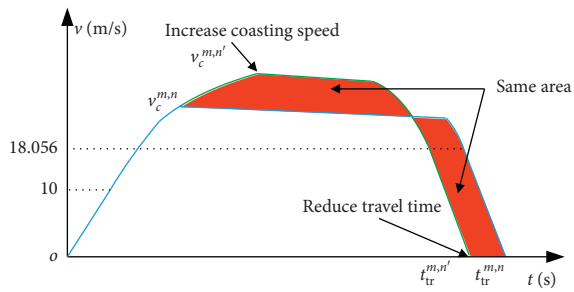


Figure 1: Relationship between coasting speed and travel time.

with the Gate Recurrent Unit (GRU) network. The MGA-GRU method consists of four stages. Specifically, in the first stage, the optimal energy timetable without a disturbance is produced by MGA. In the second stage, the dwell time and coasting speed of each train are used as decision variables. And MGA is used to provide effective actions under different disturbances, which are used as the training set. In the third stage, the outcomes of MGA are used to train the GRU network. All the above three stages are offline. In the fourth stage, the well-trained GRU network is used as a decision network. The well-trained decision network can provide effective solutions in real time after a disturbance occurs. This stage is real-time.

3.1. Modified Genetic Algorithm. In the first stage, the energy optimization objective function without a disturbance can be expressed as

$$\begin{cases} \min\ E(t) = \sum_{m=1}^{M}\sum_{n=2}^{N}\int_0^{t_{tr}^{m,n}} \left(P_T^{m,n}(t) - P_F^{m,n}(t)\right)\mathrm{d}t \\ \quad m \in (1,2,\cdots,M), n \in (2,\cdots,N), \\ \text{s.t.}\quad t_{de}^{m,n+1} = t_{de}^{m,n} + t_{tr}^{m,n} + t_{dw}^{m,n}, \\ \quad t_{tr}^{m,n} = f\left(v_c^{m,n}\right), \\ \quad 18\ \mathrm{m/s} \le v_c^{m,n} \le 22\ \mathrm{m/s}. \end{cases} \tag{9}$$

In the second stage, if a disturbance occurs at station no. n_0 of train no. m_0, the energy optimization objective function under a disturbance can be expressed as

$$\begin{cases} \min\ E(t,\varepsilon) = \sum_{m=1}^{M}\sum_{n=2}^{N}\int_0^{t_{tr}^{m,n}} \left(P_T^{m,n}(t,\varepsilon) - P_F^{m,n}(t,\varepsilon)\right)\mathrm{d}t \\ \quad m \in (1,2,\cdots,M), n \in (2,\cdots,N), \\ \text{s.t.}\quad t_{de}^{m,n+1} = \begin{cases} t_{de}^{m,n} + t_{tr}^{m,n} + t_{dw}^{m,n} + \varepsilon, & \text{if } m = m_0, n = n_0, \\ t_{de}^{m,n} + t_{tr}^{m,n} + t_{dw}^{m,n}, & \text{others}, \end{cases} \\ \quad t_{tr}^{m,n} = f\left(v_c^{m,n}\right), \\ \quad 18\ \mathrm{m/s} \le v_c^{m,n} \le 22\ \mathrm{m/s}, \end{cases} \tag{10}$$

where ε is a nonzero random variable.

Equations (9) and (10) are single-objective optimization problems that can be solved by Genetic Algorithm (GA). However, using general GA to solve such complex optimization problems, the solution is easy to fall into local optimal rather than global optimal. In order to overcome the problem of premature convergence of general GA, this paper introduces a modified Genetic Algorithm (MGA) based on Simulated Annealing (SA) algorithm to avoid falling into local optimum and approach global optimum.

Pseudocode for MGA is provided in Algorithm 1. GEN is the generation of MGA, and GEN_MAX is the maximum generation of MGA. A is the number of initial individuals, and B is the maximum number of local searches per individual. $k_c\,(0 \le k_c \le 1)$ and $k_m\,(0 \le k_m \le 1)$ are random values. $\mathrm{FIT}_{\mathrm{ind}}(\alpha)$ is the fitness of the individual α, and $\mathrm{FIT}_{\mathrm{nei}}(\alpha)$ is the fitness of the best neighborhood solution of the individual α. Each individual α contains a series of coasting speed (V) and dwell time (T). When adopting V and T, the net energy consumption is equal to $\mathrm{FIT}(\alpha)$. So, if no dwell

disturbance occurs, $\mathrm{FIT}(\alpha)$ can be calculated based on equation (9), which means that $\mathrm{FIT}(\alpha) = E(t)$. And if a dwell disturbance occurs, $\mathrm{FIT}(\alpha)$ can be calculated based on equation (10), which means that $\mathrm{FIT}(\alpha) = E(t, \varepsilon)$.

3.2. Gate Recurrent Unit. The Gate Recurrent Unit (GRU) network belongs to one of the Recurrent Neural Networks (RNN). Like the Long Short-Term Memory (LSTM) network, GRU is also proposed to solve the problems of long-term memory and gradient in backpropagation. Better than LSTM, because GRU has fewer parameters, the training speed is faster, and less data is required to generalize. While using GRU can achieve the same effect as LSTM, the GRU network is easier to be trained and the training efficiency is higher. The GRU network has a strong generalization ability and has been successfully and widely used in voice recognition, computer vision, and other fields. The structure and application of GRU are introduced below.

3.2.1. Input and Output Structure of GRU. The input and output structure of GRU is the same as the Naïve RNN, as shown in Figure 2. There is a current input x^t and the hidden state h^{t-1} passed from the previous node. The hidden state h^{t-1} contains information about the previous node. Combining with x^t and h^{t-1}, GRU produces the output y^t of the current hidden node and the hidden state h^t passed to the next node.

3.2.2. Internal Structure of GRU. The states of the two gates (r^t and z^t) are obtained by the hidden state h^{t-1} passed from the previous node and the input x^t of the current node. As shown in equations (11) and (12), r^t is a reset gate that controls reset, and z^t is an update gate that controls update. And σ is the sigmoid function. With this function, r^t and z^t can be transformed into the range $[0, 1]$, which can be used as a gating signal. W^{xr}, W^{xz}, W^{xh} and W^{hr}, W^{hz}, W^{hh} denote weight matrices of the reset gate, the update gate, and the hidden layer, respectively. b^r, b^z, b^h are the bias matrices.

$$r^t = \sigma\left(W^{xr}x^t + W^{hr}h^{t-1} + b^r\right), \tag{11}$$

$$z^t = \sigma\left(W^{xz}x^t + W^{hz}h^{t-1} + b^z\right). \tag{12}$$

After obtaining the gating signal, the reset gate is the first to be used to produce the reset data $h^{t-1'} = h^{t-1} \odot r^t$. Then $h^{t-1'}$ is stitched with the input x^t. A tanh function is used to shrink the data to the range $[-1, 1]$, that is, $h^{t'}$, as shown in equation (13). $h^{t'}$ mainly contains the current input x^t. Adding $h^{t'}$ to the current hidden state in a targeted manner is equivalent to remembering the current state.

$$h^{t'} = \tanh\left(W^{xh}x^t + W^{hh}h^{t-1'} + b^h\right). \tag{13}$$

In the update memory stage, two steps of forgetting and memorizing are performed at the same time. The expression is as follows:

$$h^t = \left(1 - z^t\right) \odot h^{t-1} + z^t \odot h^{t'}, \tag{14}$$

where $\odot$ is the Hadamard product, which is to multiply the corresponding elements in the matrix. $\oplus$ represents the matrix addition operation. The range of the gating signal z^t is $[0, 1]$. The closer the gating signal is to 1, the more data is remembered; the closer it is to 0, the more is forgotten. $(1 - z^t) \odot h^{t-1}$ means to selectively forget the previous hidden state, that is, to forget some unimportant information in the dimension of h^{t-1}. $z^t \odot h^{t'}$ indicates selective memory of the current node information $h^{t'}$. It can be seen from equation (14) that the same update gate z^t can be used for forgetting and selectively memorizing, while LSTM needs to use multiple gates. The model parameters including all $W^{xr}, W^{xz}, W^{xh}, W^{hr}, W^{hz}, W^{hh}, b^r, b^z, b^h$ are shared by all time steps and learned during model training.

In summary, the internal structure of GRU is shown in Figure 3.

3.2.3. Application of GRU. Each decision under different disturbances can be produced by MGA, which includes a series of coasting speeds and dwell time. The outcomes of MGA can be used to train the GRU network, that is, the decision network. A well-trained decision network can provide intelligent decisions in real time after a random disturbance occurs. Figure 4 shows the structure of the coasting speed and dwell time decision network. The decision network consists of five parts: input layers, previous hidden layer, current hidden layer, a decision network, and a voter. After a disturbance occurs, at the departure instant, the decision network determines the coasting speed of each departing train and the dwell time at the next station. The speed, position, and driving state of other trains, along with the train number and station number of the departing train, are put into the input layer. The train number and station number of the departing train correspond to one GRU cell. Finally, the voter gives the coasting speed and dwell time of the departing train. Furthermore, for a metro system with M trains, the input layer has $3M - 1$ neurons, and the output layer has 2 neurons.

The dwell disturbance is used in the input layers of the GRU network. After a dwell disturbance occurs at the disturbed trains, the position, speed, and driving state of the disturbed train are delayed as well. Therefore, when a train departs from a station, the position, speed, and driving state of the disturbed train (as one of the other trains) are different from the situation where no dwell disturbance occurs. In a word, the dwell disturbance influences the disturbed train's the position, speed, and driving state at each departing instant, which are used in the input layers of the GRU network.

4. Experimental Verification

In order to verify the energy-saving effect and real-time performance of the proposed MGA-GRU method for solving the TTR problem, four numerical experiments are conducted in this section. In experiment 1, MGA is used to

Randomly generate the first generation. The population contains many individuals, and each individual contains a series of decision variables: coasting speed (**V**), dwell time (**T**). Ω is used as a series of decision variables, which means $\Omega = \{\mathbf{V}, \mathbf{T}\}$. GEN $\longleftarrow$ 1.
while GEN $\in \{1, \cdots, \text{GEN\_MAX}\}$ **do**
  Initialize the neighborhood structures set $\Omega(\alpha, \beta)$.
  $\alpha \longleftarrow 1$, $\beta \longleftarrow 1$.
  while $\alpha \in \{1, \cdots, A\}$ **do**
    while $\beta \in \{1, \cdots, B\}$ **do**
      Randomly generate a series of **V** and **T** within constraints to serve as a set of solutions $\Omega(\alpha, \beta)$ in the neighborhood of $\Omega(\alpha)$.
      $\beta \longleftarrow \beta + 1$.
    end while
    $\alpha \longleftarrow \alpha + 1$, $\beta \longleftarrow 1$.
  end while
  Equations (9) and (10) are used to evaluate the fitness $\text{FIT}(\alpha, \beta)$ of the solution $\Omega(\alpha, \beta)$ without a disturbance and under a disturbance, respectively.
  if $\text{FIT}_{\text{nei}}(\alpha) < \text{FIT}_{\text{ind}}(\alpha)$ **then**
    $\text{FIT}_{\text{ind}}(\alpha) = \text{FIT}_{\text{nei}}(\alpha)$.
  else if $e^{[\text{FIT}_{\text{ind}}(\alpha) - \text{FIT}_{\text{nei}}(\alpha)]/\text{Temper}} < \text{Rand}(0, 1)$ **then**
    $\text{FIT}_{\text{ind}}(\alpha) = \text{FIT}_{\text{nei}}(\alpha)$.
  end if
  GEN $\longleftarrow$ GEN + 1.
  Decrease Temper.
  if the probability of crossover $P_c \geq k_c$ **then**
    Crossover.
  end if
  if the probability of mutation $P_m \geq k_m$ **then**
    Mutation.
  end if
end while
Output the global optimum.

Algorithm 1: Modified Genetic Algorithm (MGA).

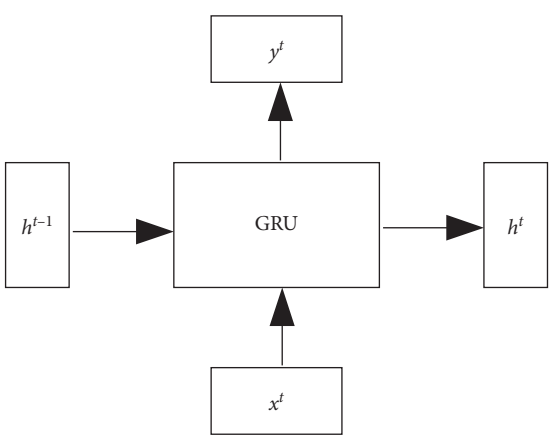


Figure 2: Input and output structure of GRU.

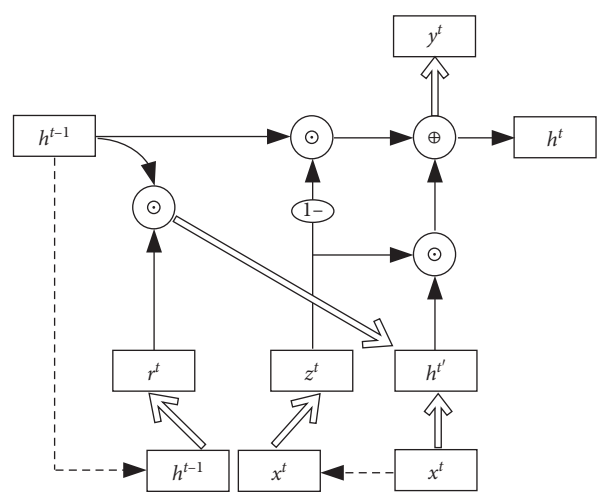


Figure 3: The internal structure of GRU.

produce the optimal timetable without a disturbance in the two-train metro system. In experiment 2, the timetable is rescheduled after a disturbance occurs, by using the MGA-GRU method. In experiment 3, the MGA-GRU method is applied for solving the TTR problem in a three-train metro system. In experiment 4, the MGA-GRU method is applied in a bidirectional metro system with five trains on two tracks.

The information of the pilot metro system is shown in Figure 5. The configuration of the numerical experiment is shown in Table 1.

Some settings for the four experiments are listed in Table 2. Based on the above settings, there are no traffic jam

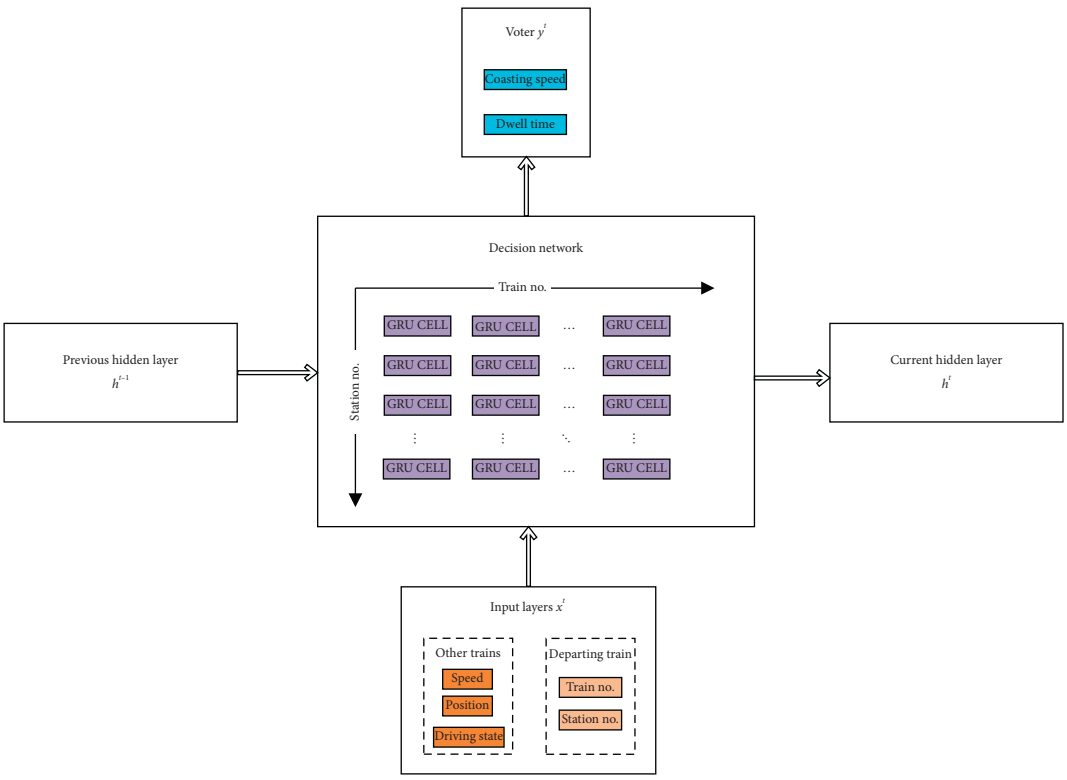


Figure 4: Structure of decision network.

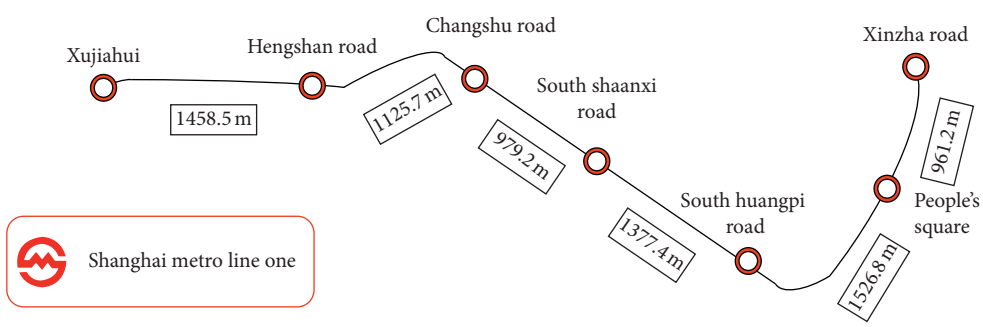


Figure 5: Information of the pilot metro system.

Table 1: Configuration of numerical experiment.

Items	Configuration information
Operating system	Windows 10
CPU	Intel® Xeon® CPU E5-2620 v4 @ 2.10 GHz
RAM	32 GB
Simulation environment	TensorFlow
Locales	Python

Table 2: Settings for the four experiments.

Items	Range
The headway	120 s
Dwell time	[30, 35] s
Dwell disturbance	[−5, 15] s
Coasting speed	[18, 22] m/s
η_1	0.8
η_2	0.7
η_3	0.7

conflicts between trains. In experiments 2 and 3, the dwell disturbance occurs at train no. 1 at Changshu Road Station, while in experiment 4, the dwell disturbance can occur at train no. 1 at Hengshan Road Station or at Changshu Road Station or at South Shaanxi Road Station.

4.1. Experiment 1. The goal of experiment 1 is to validate MGA for solving the energy optimization problem and produce the optimal timetable without a disturbance. MGA is used to solve the energy optimization problem based on equation (7) and the optimal timetable as is shown in Table 3.

Figure 6 shows the distribution of individuals of the 1st, 10th, and 15th generations produced by MGA. As shown in Figure 6, the distribution of individuals in the 1st generation is discrete. After 10 generations, the individuals' distribution gradually concentrates. Finally, after 15 generations, the individuals converge with a fitness value of 256.02 kWh, which is the optimal energy consumption without a disturbance.

In order to prove the effectiveness of MGA, a general GA is also applied to produce the offline timetable without a disturbance under the same condition. Figure 7 shows the distribution of individuals of the 1st, 10th, and 15th generations produced by a general GA. After 15 generations, the individuals converge with a fitness value of 276.59 kWh. It can be seen from Figure 7 that the individuals of a general GA concentrate more quickly than those of MGA. What is more, the fitness of a general GA (276.59 kWh) is bigger than that of MGA (256.02 kWh). Therefore, compared with a general GA, the proposed MGA can avoid falling into local optimum prematurely and provide a better solution.

4.2. Experiment 2. This experiment is to use the MGA-GUR method to solve the TTR problem under random disturbances in the two-train metro system. The energy-saving effect of the MGA-GRU method is reflected by the saved energy during test compared with the no-action strategy. The real-time performance of the MGA-GRU method is reflected by the time it takes to provide a pair of strategies during testing.

4.2.1. Dataset. The outputs of MGA based on equation (8) under the disturbance from $(10\,\text{s}, 10.2\,\text{s}, 10.4\,\text{s}, \ldots, 14.8\,\text{s}, 15.0\,\text{s})$ are selected as the dataset to train the MGA-GRU network.

4.2.2. Baselines. The proposed MGA-GRU method is compared against two baselines: (i) *no action*, where each train does not take any measures after disturbances occur and (ii) *MGA*, where MGA is used to give an offline strategy to deal with the dwell disturbances. The two baselines are representative of the worst and best possible strategies. It is expected that MGA-GRU falls in between these two extreme cases. It should be emphasized that MGA-GRU can reschedule the timetable in real time.

The changes of the net energy consumption with these three strategies are compared during testing. The dwell disturbance which occurs at train no. 1 at Changshu Road Station is 13.45 s. Figure 8 shows the net energy consumption curve of the whole journey using the three methods. It can be seen from Figure 8 that from the moment when the disturbance occurs, the net energy consumption of no action has always remained the highest. Although MGA can achieve good results in saving energy, it takes a lot of time to provide a decision, which does not meet the real-time requirements for solving the TTR problem. Compared with the above two methods, MGA-GRU can reschedule the timetable in real time and achieve saving energy.

The rescheduled timetable with the MGA-GRU method under a 13.45 s dwell disturbance in the two-train metro system is shown in Table 4.

Table 5 shows the average calculation time and average total energy consumption of the three strategies in 10 tests. The MGA-GRU strategy is energy efficient compared with the no-action strategy and its average calculation time is only 0.15 s which meets the requirements of real-time effect. Therefore, MGA-GRU can reschedule the timetable in real time and achieve saving energy after a dwell disturbance occurs. It should be noted that although the total energy consumption of the MGA strategy is less than the MGA-GRU strategy, it requires a greatly long calculation time (8694.08 s in total) to reschedule the timetable, which absolutely does not meet the real-time requirements of the TTR problem. In terms of calculation time, MGA-GRU has an absolute superiority.

4.3. Experiment 3. In the real case of SML1, there are at most three trains between two substations. Therefore, it is essential to apply the MGA-GRU method to a three-train metro system. What is more, according to the real case of SML1 [35], the train departs early frequently, which means that the value of a disturbance can be negative. This situation is also taken into consideration in experiment 3.

First, MGA produces the optimal timetable without a disturbance based on equation (7), as shown in Table 6.

4.3.1. Dataset. The outputs of MGA under the disturbance from $(10\,\text{s}, 10.2\,\text{s}, 10.4\,\text{s}, \ldots, 14.8\,\text{s}, 15.0\,\text{s})$ and $(-5.0\,\text{s}, -4.8\,\text{s}, \ldots, -0.4\,\text{s}, -0.2\,\text{s}, 0.0\,\text{s})$ are selected as the dataset to train the MGA-GRU network.

4.3.2. Baselines. Same as experiment 2, in the three-train metro system, MGA-GRU is also compared against two baselines: no action and MGA.

Figure 9 gives the net energy consumption curve of the whole journey under a −2.37 s dwell disturbance (departing early) in the three-train metro system. As can be seen from Figure 9, the MGA method has the best energy-saving effect. However, it also requires a long calculation time.

The rescheduled timetable with the MGA-GRU method under a −2.37 s dwell disturbance in the three-train metro system is shown in Table 7.

Table 3: The optimal timetable of the two-train system.

Section	Spacing (m)	Train no.	Departure instant	Arrival instant	Dwell time (s)	Coasting speed (m/s)
Xujiahui ⟶ Hengshan Road	1458.5	1	8:00:00	8:01:30	29.6	21.8
		2	8:02:00	8:03:42	20.1	18
Hengshan Road ⟶ Changshu Road	1125.7	1	8:01:59	8:03:21	28.4	18
		2	8:04:02	8:05:22	27.2	18.32
Changshu Road ⟶ South Shaanxi Road	979.2	1	8:03:49	8:04:56	22.0	21.08
		2	8:05:49	8:07:01	23.4	18.4
South Shaanxi Road ⟶ South Huangpi Road	1377.4	1	8:05:18	8:06:45	23.1	21.44
		2	8:07:25	8:09:01	20	18
South Huangpi Road ⟶ People's Square	1526.8	1	8:07:08	8:08:54	20	18.04
		2	8:09:21	8:11:06	26.5	18.32
People's Square ⟶ Xinzha Road	961.2	1	8:09:14	8:10:25	—	18.04
		2	8:11:32	8:12:44	—	18.04

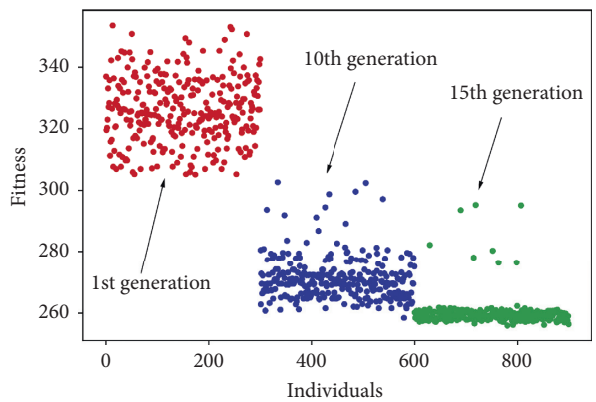


Figure 6: Individuals' distribution of the 1st, 10th, and 15th generations produced by MGA.

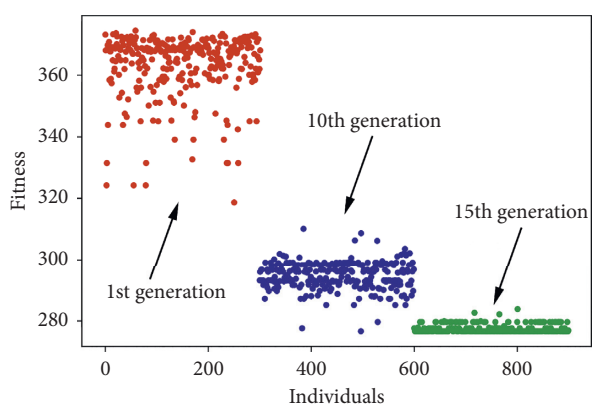


Figure 7: Individuals' distribution of the 1st, 10th, and 15th generations produced by a general GA.

Table 8 shows the average calculation time and average total energy consumption of the three strategies in 10 tests. The average calculation time of MGA-GRU to provide the coasting speed and dwell time of each group is only 0.27 s, which meets the real-time requirements of the TTR problem. Besides, the MGA-GRU strategy is energy efficient compared with the no-action strategy. Therefore, MGA-GRU can reschedule the timetable in real time and achieve energy saving after a random dwell disturbance occurs.

4.4. Experiment 4. According to the real case of SML1, there exist at most 3 trains on one track between 2 substations. The goal of experiment 4 is to apply MGA-GRU to a real metro system, which is a bidirectional metro line with five trains on

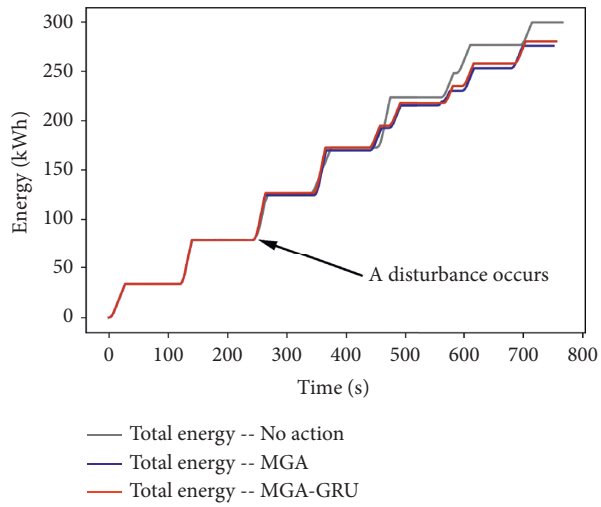


Figure 8: Changes in net energy consumption with different strategies under a 13.45 s dwell disturbance.

Table 4: The rescheduled timetable with the MGA-GRU method in the two-train system.

Section	Spacing (m)	Train no.	Departure instant	Arrival instant	Dwell time (s)	Coasting speed (m/s)
Xujiahui ⟶ Hengshan Road	1458.5	1	8:00:00	8:01:30	29.6	21.8
		2	8:02:00	8:03:35	21.2	18.04
Hengshan Road ⟶ Changshu Road	1125.7	1	8:01:59	8:02:21	41.9	18
		2	8:03:56	8:05:12	22	18
Changshu Road ⟶ South Shaanxi Road	979.2	1	8:04:03	8:05:11	29.6	18.28
		2	8:05:34	8:06:43	20	18.2
South Shaanxi Road ⟶ South Huangpi Road	1377.4	1	8:05:41	8:07:11	29.8	18
		2	8:07:03	8:08:33	20	18.08
South Huangpi Road ⟶ People's Square	1526.8	1	8:07:41	8:09:13	20	18
		2	8:08:53	8:10:26	21.8	18.04
People's Square ⟶ Xinzha Road	961.2	1	8:09:33	8:10:39	—	18.56
		2	8:10:48	8:11:55	—	18

Table 5: Performances with three strategies in the two-train metro system.

Strategy	Calculation time (s)	Net traction energy consumption (kWh)	Energy-saving percentage compared with no action strategy (%)
No action	0	293.95	—
MGA	8694.08	274.58	6.59
MGA-GRU	**0.15**	**280.88**	**4.45**

two tracks. Different from experiments 2 and 3, in experiment 4, the disturbances' range is $[-5, 15]$ s, and each dwell disturbance can occur at Hengshan Road Station or at Changshu Road Station or at South Shaanxi Road Station.

The bidirectional metro line with five trains on two tracks is shown in Figure 10. The fourteen stations are numbered in sequence from 1 to 14. There are three trains departing from station no.1 and travels in sequence to station no.7, which is called up direction. Then, there is a turning of 60 s in duration from station no.7 to station no.8. After that, each train drives from station no.8 to station no.14, which is called down direction. After that, there is also a turning of 60 s in duration from station no.14 back to station no.1. And there are other two trains departing from station no.8 to station no.7 and then back to station no.8. The departure instant of train no.1 at station no.1 and train no.4 at station no.8 is the same. Besides, the headway time of every two train is also 120 s.

First, MGA is also used to produce the optimal timetable under no disturbance.

4.4.1. Dataset. The outputs of MGA under the disturbance from $(-5.0\,\text{s}, -4.8\,\text{s}, -4.6\,\text{s}, \ldots, 14.6\,\text{s}, 14.8\,\text{s}, 15.0\,\text{s})$ are selected as the dataset to train the MGA-GRU network.

4.4.2. Baselines. In the five-train metro system, the proposed MGA-GRU method is compared against three baselines: (i) *no action*, where each train does not take any measures after disturbances occur; (ii) *general GA*, where a general GA is used to give an offline strategy to deal with the dwell disturbances; and (iii) *MGA*, where MGA is used to give an

Table 6: The optimal timetable of the three-train system.

Section	Spacing (m)	Train no.	Departure instant	Arrival instant	Dwell time (s)	Coasting speed (m/s)
Xujiahui ⟶ Hengshan Road	1458.5	1	8:00:00	8:01:41	20	18.12
		2	8:02:00	8:03:29	20	21.96
		3	8:04:00	8:05:42	25.3	18
Hengshan Road ⟶ Changshu Road	1125.7	1	8:02:01	8:03:18	20	19.88
		2	8:03:49	8:05:05	20.2	20.52
		3	8:06:07	8:07:25	29.5	19.16
Changshu Road ⟶ South Shaanxi Road	979.2	1	8:03:38	8:04:51	26.3	18
		2	8:05:25	8:06:32	27.5	21.32
		3	8:07:55	8:09:07	26.8	18
South Shaanxi Road ⟶ South Huangpi Road	1377.4	1	8:05:17	8:06:54	20	18
		2	8:07:00	8:08:26	20.1	21.52
		3	8:09:34	8:11:00	20	21.84
South Huangpi Road ⟶ People's Square	1526.8	1	8:07:14	8:09:00	20	18.04
		2	8:08:46	8:10:20	20	21.4
		3	8:11:20	8:13:06	20	18.04
People's Square ⟶ Xinzha Road	961.2	1	8:09:20	8:10:25	—	21.6
		2	8:10:40	8:11:47	—	21.12
		3	8:13:26	8:14:33	—	20.08

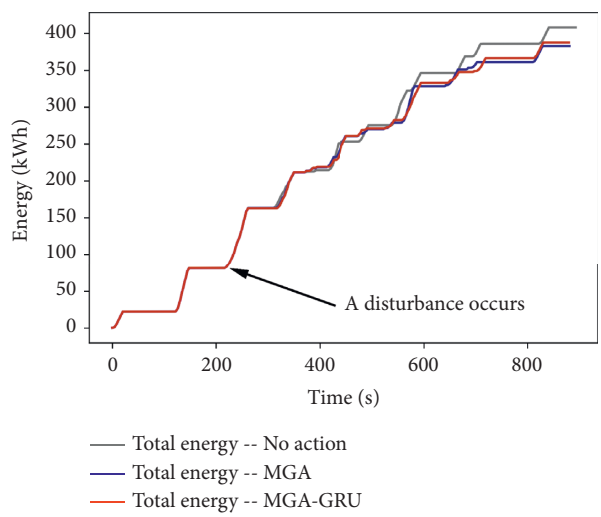


Figure 9: Changes in net energy consumption with different strategies under the −2.37 s dwell disturbance.

offline strategy to deal with the dwell disturbances. It is expected that MGA-GRU can achieve better energy saving effect than a general GA, which can prove the effectiveness of MGA.

Table 9 shows the average calculation time and average total energy consumption of the four strategies in 20 tests. The average calculation time of MGA-GRU to provide the coasting speed and dwell time of each group is only 0.33 s, which meets the real-time requirements of the TTR problem as well. And compared with no-action strategy, the MGA-GRU strategy is energy efficient compared with the no-action strategy and can save an average of 7.19% energy. Therefore, MGA-GRU can reschedule the timetable in real time and achieve energy saving after a random dwell disturbance occurs. What is more, MGA (8.73%) achieves better energy saving effect than a general GA (6.15%), which also proves the effectiveness of MGA.

The decision time of MGA-GRU with the onboard computer's configuration is also discussed. And the configuration of the train's onboard computer is shown in Table 10.

The configuration on the PC is restricted to the same as the train's onboard computer, and then experiment 4 is performed again. The experimental results show that the average calculation time is 0.35 s, which reflects that the proposed MGA-GRU method can be applied to the onboard computer.

Table 7: The rescheduled timetable with the MGA-GRU method of the three-train system.

Section	Spacing (m)	Train no.	Departure instant	Arrival instant	Dwell time (s)	Coasting speed (m/s)
Xujiahui ⟶ Hengshan Road	1458.5	1	8:00:00	8:01:41	20	18.12
		2	8:02:00	8:03:29	29.9	19.88
		3	8:04:00	8:05:35	26.8	18
Hengshan Road ⟶ Changshu Road	1125.7	1	8:02:01	8:03:18	17.6	21.8
		2	8:03:59	8:05:13	20	20.92
		3	8:06:01	8:07:15	20.1	21.84
Changshu Road ⟶ South Shaanxi Road	979.2	1	8:03:35	8:04:42	27.3	20.68
		2	8:05:33	8:06:39	28.1	20.76
		3	8:07:35	8:08:41	20	18.76
South Shaanxi Road ⟶ South Huangpi Road	1377.4	1	8:05:09	8:06:35	24	18.76
		2	8:07:07	8:08:32	21.5	19.92
		3	8:09:01	8:10:26	20.1	21.76
South Huangpi Road ⟶ People's Square	1526.8	1	8:06:59	8:08:31	21.7	19.52
		2	8:08:54	8:10:27	25.8	19
		3	8:10:47	8:12:19	22.8	19.08
People's Square ⟶ Xinzha Road	961.2	1	8:08:53	8:09:59	—	18.2
		2	8:10:52	8:11:58	—	20.56
		3	8:12:42	8:13:47	—	18

Table 8: Performances with three strategies in the three-train metro system.

Strategy	Calculation time (s)	Net traction energy consumption (kWh)	Energy-saving percentage compared with no-action strategy (%)
No action	0	412.23	—
MGA	13768.68	380.82	7.62
MGA-GRU	**0.27**	**386.84**	**6.16**

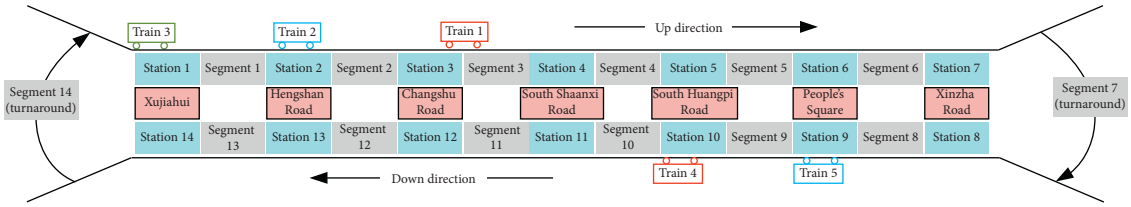


Figure 10: Bidirectional metro line with five trains on two tracks.

Table 9: Performances with three strategies in the five-train metro system.

Strategy	Calculation time (s)	Net traction energy consumption (kWh)	Energy-saving percentage compared with no-action strategy (%)
No action	0	741.85	—
General GA	17941.36	696.22	6.15
MGA	21074.15	677.09	8.73
MGA-GRU	**0.33**	**688.51**	**7.19**

Table 10: Configuration of the train's onboard computer.

Items	Configuration information
CPU	Intel Celeron G3900
RAM	4 GB DDR4 2133 MHz
ROM	120G SSD

5. Conclusion

In this paper, a Modified Genetic Algorithm-Gate Recurrent Unit (MGA-GRU) method is proposed to solve the train timetable rescheduling (TTR) problem. The proposed MGA-GRU method can reschedule timetable to optimize the net traction energy consumption of the metro system under a random dwell disturbance in real time. Specifically, the outcomes of modified Genetic Algorithm (MGA) under different dwell disturbances are used as the training set to train the Gate Recurrent Unit (GRU) network (the decision network). After a disturbance occurs, the well-trained decision network can provide appropriate coasting speed and dwell time in real time.

Better than traditional optimization methods, such as enhanced brute force (EBF), ant colony optimization (ACO), and Genetic Algorithm (GA), MGA-GRU can achieve real-time train timetable rescheduling. Superior to CDSA, MGA-GRU can rearrange the coasting speed and dwell time of all trains in real time after disturbances occur, so as to achieve better energy-saving effect.

Four experiments are conducted on the Shanghai Metro Line One (SML1) pilot network to verify the energy-saving effect and real-time performance of the proposed method. The experimental results show that in the two-train metro system, the three-train metro system, and the five-train metro system (a bidirectional metro line on two tracks) after a disturbance occurs, the MGA-GRU strategy can save an average of 4.45%, 6.16%, and 7.19% of energy compared with the no-action strategy, while the average calculation time for each group of coasting speed and dwell time is only 0.15 s, 0.27 s, and 0.33 s, respectively. In all the two-train metro system, the three-train metro system, and the five-train metro system, the proposed MGA-GRU method can solve the TTR problem under random disturbances in real time.

In the future work, according to Taguchi's experimental design method [36] and other intelligent optimization methods [37], the impact of user-defined parameters on the performance of the proposed algorithm should be analyzed, other parameters should be compared, and the best settings should be decided.

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 regarding the publication of this paper.

Acknowledgments

The authors would like to acknowledge the fund provided by the Shanghai Shentong Metro Group Co., Ltd., which also provided experimental environment and assistance. This study was funded by the Shanghai Shentong Metro Group Co., Ltd. (Grant no. JS-372 KY09R013).

References

[1] D. Liu, S. Zhu, Y. Bi, K. Liu, and Y. Xu, "Research on the utilization of metro regenerative braking energy based on an improved differential evolution algorithm," *Journal of Advanced Transportation*, vol. 2020, Article ID 7085809, 11 pages, 2020.

[2] S. Su, T. Tang, and C. Roberts, "A cooperative train control model for energy saving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 622–631, 2015.

[3] J. C. Jong and E. F. Chang, "Models for estimating energy consumption of electric trains," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 6, pp. 278–291, 2005.

[4] Z. Luo, X. Li, and N. Xiu, "A sparse optimization approach for energy-efficient timetabling in metro railway systems," *Journal of Advanced Transportation*, vol. 2018, Article ID 1784789, 19 pages, 2018.

[5] Y. Zhou, Y. Bai, J. Li, B. Mao, and T. Li, "Integrated optimization on train control and timetable to minimize net energy consumption of metro lines," *Journal of Advanced Transportation*, vol. 2018, Article ID 7905820, 19 pages, 2018.

[6] H. Liu, M. Zhou, X. Guo, Z. Zhang, B. Ning, and T. Tang, "Timetable optimization for regenerative energy utilization in subway systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3247–3257, 2019.

[7] A. Czerwiński, S. Obrębowski, and Z. Rogulski, "New high-energy lead-acid battery with reticulated vitreous carbon as a carrier and current collector," *Journal of Power Sources*, vol. 198, pp. 378–382, 2012.

[8] A. Burke and M. Miller, "The power capability of ultracapacitors and lithium batteries for electric and hybrid vehicle applications," *Journal of Power Sources*, vol. 196, no. 1, pp. 514–522, 2011.

[9] Y. Suzuki, A. Koyanagi, M. Kobayashi, and R. Shimada, "Novel applications of the flywheel energy storage system," *Energy*, vol. 30, no. 11-12, pp. 2128–2143, 2005.

[10] R. Mukherjee, R. Krishnan, T.-M. Lu, and N. Koratkar, "Nanostructured electrodes for high-power lithium ion batteries," *Nano Energy*, vol. 1, no. 4, pp. 518–533, 2012.

[11] P. Liu, L. Yang, Z. Gao, Y. Huang, S. Li, and Y. Gao, "Energy-efficient train timetable optimization in the subway system with energy storage devices," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3947–3963, 2018.

[12] T. Albrecht, "Train running time control using genetic algorithms for the minimization of energy costs in DC rapid transit systems," *Challenges in Real World Optimisation Using Evolutionary Computing*, vol. 1, Dresden University of Technology, Faculty of Traffic Sciences, Dresden, Germany.

[13] X. Meng, L. Jia, and W. Xiang, "Complex network model for railway timetable stability optimisation," *IET Intelligent Transport Systems*, vol. 12, no. 10, pp. 1369–1377, 2018.

[14] J. Yin, D. Chen, L. Yang, T. Tang, and B. Ran, "Efficient real-time train operation algorithms with uncertain passenger demands," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2600–2612, 2016.

[15] A. González-Gil, R. Palacin, and P. Batty, "Sustainable urban rail systems: strategies and technologies for optimal management of regenerative braking energy," *Energy Conversion and Management*, vol. 75, pp. 374–388, 2013.

[16] C. S. Chang and S. S. Sim, "Optimising train movements through coast control using genetic algorithms," *IEE Proceedings—Electric Power Applications*, vol. 144, no. 1, pp. 65–73, 1997.

[17] B. R. Ke and N. Chen, "Signalling blocklayout and strategy of train operation for saving energy in mass rapid transit systems," *IEE Proceedings—Electric Power Applications*, vol. 152, no. 2, pp. 129–140, 2005.

[18] X. Meng, L. Jia, Y. Qin, J. Xu, and T. Zhou, "Study on train operation adjustment based on hybrid convergent particle swarm optimization," in *Proceedings of the 2009 International Conference on Measuring Technology and Mechatronics Automation*, vol. 3, pp. 326–329, Hunan, China, April 2009.

[19] X. Yang, B. Ning, X. Li, and T. Tang, "A two-objective timetable optimization model in subway systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1913–1921, 2014.

[20] F. A. Ortega, M. A. Pozo, and J. Puerto, "On-line timetable rescheduling in a transit line," *Transportation Science*, vol. 52, no. 5, pp. 1106–1121, 2018.

[21] P. Xu, F. Corman, Q. Peng, and X. Luan, "A timetable rescheduling approach and transition phases for high-speed railway traffic during disruptions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2607, no. 1, pp. 82–92, 2017.

[22] D. Šemrov, R. Marsetič, M. Žura, L. Todorovski, and A. Srdic, "Reinforcement learning approach for train rescheduling on a single-track railway," *Transportation Research Part B: Methodological*, vol. 86, pp. 250–267, 2016.

[23] Z. Hou, H. Dong, S. Gao, G. Nicholson, L. Chen, and C. Roberts, "Energy-saving metro train timetable rescheduling model considering ATO profiles and dynamic passenger flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2774–2785, 2019.

[24] N. Zhao, C. Roberts, S. Hillmansen, and G. Nicholson, "A multiple train trajectory optimization to minimize energy consumption and delay," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2363–2372, 2015.

[25] C. Gong, S. Zhang, F. Zhang, J. Jiang, and X. Wang, "An integrated energy-efficient operation methodology for metro systems based on a real case of Shanghai metro line one," *Energies*, vol. 7, no. 11, pp. 7305–7329, 2014.

[26] X. Zuo, C. Chen, W. Tan, and M. Zhou, "Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1030–1041, 2015.

[27] 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.

[28] 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.

[29] Y.-Y. Tan, Y.-X. Li, and R.-X. Wang, "Scheduling extra train paths into cyclic timetable based on the genetic algorithm," *IEEE Access*, vol. 8, pp. 102199–102211, 2020.

[30] M.-Y. Gao, N. Zhang, S.-L. Shen, and A. Zhou, "Real-time dynamic earth-pressure regulation model for shield tunneling by integrating GRU deep learning method with GA optimization," *IEEE Access*, vol. 8, pp. 64310–64323, 2020.

[31] Y. Zhang, B. Xu, and T. Zhao, "Convolutional multi-head self-attention on memory for aspect sentiment classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 4, pp. 1038–1044, 2020.

[32] S. Su, X. Li, T. Tang, and Z. Gao, "A subway train timetable optimization approach based on energy-efficient operation strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 883–893, 2013.

[33] G. Yang, F. Zhang, C. Gong, and S. Zhang, "Application of a deep deterministic policy gradient algorithm for energy-aimed timetable rescheduling problem," *Energies*, vol. 12, no. 18, p. 3461, 2019.

[34] W. J. Davis, *The Tractive Resistance of Electric Locomotives and Cars*, General Electric, Boston, MA, USA, 1926.

[35] F. Liu, R. Xu, W. Fan, and Z. Jiang, "Data analytics approach for train timetable performance measures using automatic train supervision data," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 568–577, 2018.

[36] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "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, 2019.

[37] J. Wang and T. Kumbasar, "Parameter optimization of interval type-2 fuzzy neural networks based on PSO and BBBC methods," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 247–257, 2019.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8870211, 8 pages
https://doi.org/10.1155/2020/8870211

Research Article

A Framework for Detecting Vehicle Occupancy Based on the Occupant Labeling Method

Jooyoung Lee,[1] **Jihye Byun**,[2] **Jaedeok Lim**,[3] **and Jaeyun Lee**[3]

[1]*The Cho Chun Shik Graduate School of Green Transportation, Korea Advanced Institute of Science and Technology, Daejeon 34051, Republic of Korea*
[2]*Center for Eco-Friendly Smart Vehicle, Korea Advanced Institute of Science and Technology, Daejeon 34051, Republic of Korea*
[3]*Technical Research Center, GnT Solution, Inc., Seoul 07255, Republic of Korea*

Correspondence should be addressed to Jihye Byun; snowflower@kaist.ac.kr

Received 24 August 2020; Revised 31 October 2020; Accepted 17 November 2020; Published 3 December 2020

Academic Editor: Ladislav Routil

High-occupancy vehicle (HOV) lanes or congestion toll discount policies are in place to encourage multipassenger vehicles. However, vehicle occupancy detection, essential for implementing such policies, is based on a labor-intensive manual method. To solve this problem, several studies and some companies have tried to develop an automated detection system. Due to the difficulties of the image treatment process, those systems had limitations. This study overcomes these limits and proposes an overall framework for an algorithm that effectively detects occupants in vehicles using photographic data. Particularly, we apply a new data labeling method that enables highly accurate occupant detection even with a small amount of data. The new labeling method directly labels the number of occupants instead of performing face or human labeling. The human labeling, used in existing research, and occupant labeling, this study suggested, are compared to verify the contribution of this labeling method. As a result, the presented model's detection accuracy is 99% for the binary case (2 or 3 occupants or not) and 91% for the counting case (the exact number of occupants), which is higher than the previously studied models' accuracy. Basically, this system is developed for the two-sided camera, left and right, but only a single side, right, can detect the occupancy. The single side image accuracy is 99% for the binary case and 87% for the counting case. These rates of detection are also better than existing labeling.

1. Introduction

As the vehicle supply increases, the road infrastructure capacity is relatively reduced, so continuous construction of new roads is needed in many areas around the globe. However, increasing the road infrastructure capacity by building more roads is costly and time-consuming, so there is a limit that cannot accommodate the vehicle growth rate. In order to solve this problem, some policies have been implemented to encourage carpooling, such as reducing travel time through HOV lanes or providing discounts on congestion tolls from multipassenger vehicles [1]. To enforce this policy, technology for detecting vehicle occupants is essential. Currently, when enforcing HOV lane control policies or providing congestion toll discounts to multipassenger vehicles, employees visually estimate the number of passengers in each vehicle by checking the video data in management centers [2]. This manual method is labor-intensive, lowers operational efficiency, and increases labor costs. In the United States, which is cracking down on the illegal use of HOV lanes, the actual violation rate is about 50–80%, but the crackdown rate is reported to be less than 10% [3]. In South Korea, where discounts on congestion tolls are provided, congestion is likely to increase even more during peak hours due to inspection of the number of passengers in each vehicle at the toll gates and the collection of the tolls.

To solve this problem, various studies were conducted to automate the vehicle occupant estimate process. The research can be divided into two detection technology areas: using in-vehicle sensors [4–10] and using the image data from outside cameras [11–17]. When using in-vehicle sensors, the accuracy is generally high; however, all vehicles

need to be equipped with devices that can detect the number of passengers. Such devices usually use video cameras, which causes privacy concerns for many people. Therefore, the use of this method is impractical. Moreover, most studies that detect occupants using outside cameras had limited scope. For example, they can only detect the number of passengers in the front seat [12–14], only count the number of children onboard [16], or only determine if two or more passengers have boarded a vehicle. In particular, in [17], an 88% detection accuracy was achieved using image data captured outside the vehicle by one front and one side camera. This accuracy level is applicable to the real world, so pilot services were performed in several regions in the United States.

In the vehicle occupant detection field, there is another limitation in that only newly acquired images can be used as training data. Therefore, an algorithm is needed to achieve a high detection rate even with a small data set. In previous studies, a two-stage detection algorithm was used to overcome this limitation. Generally, the two-stage detection algorithm first detects the window area in the vehicle images and then detects the number of passengers in the window area only [15]. However, this algorithm has some limitations due to its complicated learning process and the increased network size, which increases the required calculation times.

Therefore, this study proposes an overall algorithmic framework that effectively detects vehicle occupants using left and right side photographic data from the vehicle exterior in a one-step process using a small amount of data. Specifically, we present a new data labeling method to accurately detect the number of occupants. The new labeling method directly labels the number of occupants instead of performing face or human labeling, which is a widely used method for image detection. Based on this advanced labeling method, this study contains only a single-stage detection algorithm. A decrease in the detection stage shrinks the network size, number of samples, and detection time.

The structure of this paper is as follows: the second section introduces an image acquisition system for detecting in-vehicle occupants and describes a new occupancy labeling method and acquired image data set; the third section describes the structure of the deep neural network used to detect occupants; the fourth section presents a discussion of the results of the presented algorithm in this study; and the final section summarizes the conclusions and implications of this study.

2. Image Acquisition and New Occupancy Labeling Method

Two infrared ray cameras, infrared ray illuminators, and a laser trigger acquire the images used for training and testing. An overview of the image acquisition system is shown in Figure 1. The cameras are located on the left and right sides of the vehicle. Through various tests, the research team determined the optimal specifications of the locations, heights, and angles of the cameras [18]. The infrared ray illuminators are used to improve the images when there is not enough visible light, such as at night or when the windows of the vehicles are tinted. The laser trigger detects the vehicle's entry into the detection zone that has the cameras. When the trigger recognizes a vehicle, the infrared ray cameras take images of the left and right sides of the vehicle. Then, the cameras send the frames to the server, and the accumulated images are used for training. When detecting vehicle occupancy, the images do not need to be transmitted to the server since they are treated by the on-site system.

As mentioned in the Introduction, previous research has labeled objects, such as faces, humans, and windows, and this labeling method has some benefits: (i) the number of labeling types, as the method needs one or two kinds of labels; (ii) securing a large number of learning samples since every image has to have one or more windows and a human. However, the method needs two stages, such as finding windows and then faces or an algorithm to divide the row of occupants. It leads to more times for calculation and higher error rates. To overcome the limitation, this study adopts a new labeling methodology to determine how many people are in the front and rear passenger seats. Therefore, each image must have two labels among six kinds of labels: one person in the front seat or two people in the front seat, and 0, 1, 2, or 3 people in the rear seat, as shown in Figure 2.

3. Vehicle Occupancy Detection Methodology

Figure 3 shows the proposed methodology for detecting occupants using the proposed labeling method in this study. An independently trained occupancy detection model is used for the images on each side, and passengers in the front seat are detected from the right side. As for the detection of occupants in the rear seat, both the left and right side images are used, and the number of occupants in the rear seat is determined using the higher detection score that results from comparing the detection scores obtained from the images of both sides. After that, the numbers of occupants in the front seat and the rear seat are added to obtain the total number of occupants.

This study trained the detection model and tested the results in the MATLAB 2019b environment. We used the Faster RCNN detection method, which has a high detection accuracy, instead of a unified detection algorithm, such as Yolo or an SSD with high speed [19]. The Faster RCNN method was introduced in [20], and it can detect multiple objects in one image with high accuracy and speed. This speeds up processing the regional-based CNN algorithm proposed in [21]. Specifically, the region proposal network (RPN), which is based on a fully convolutional network, was introduced to derive the region proposals from the feature map of the input image, as it replaces the selective search, which was a bottleneck of the training process. The RPN slides a 3 x 3 spatial window on a feature map to predict the region proposals, called multiple anchors, for each window. An anchor is the bounding box of the number of occupants that need to be detected in the input image. As in the previous paper, nine combinations of three sizes (128, 256, and 512) and three ratios (2 : 1, 1 : 1, and 1 : 2) of the anchor box were used for training in this paper. The derived anchors are classified into region proposals if the IoU (Intersection

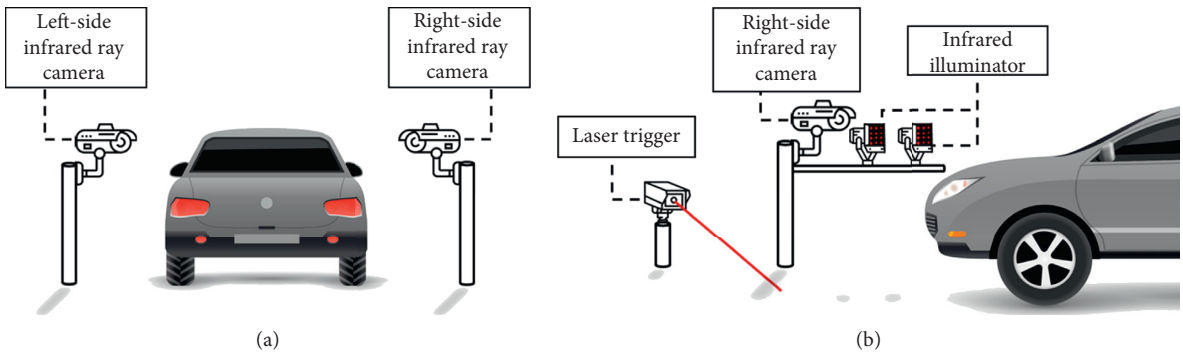


Figure 1: Overview of the image acquisition system for vehicle occupancy detection. (a) Rear view. (b) Side view.

Figure 2: Examples of the new labeling method. (a) Front seat: 1, rear seat: 1 case. (b) Front seat: 2, rear seat: 2 cases. (c) Front seat: 1, rear seat: 0 cases. (d) Front seat: 2, rear seat: 1 case.

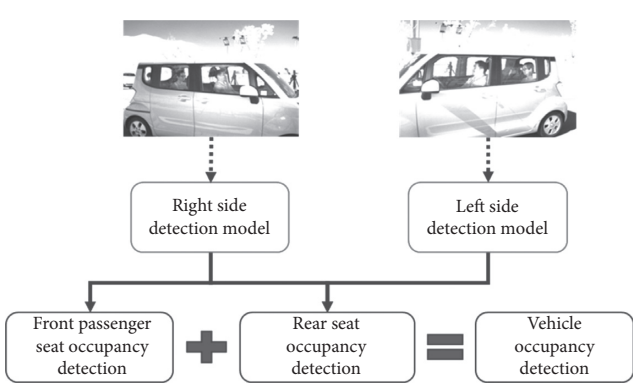


Figure 3: Overview of the vehicle occupancy detection methodology using both right and left side images.

over Union, see 1) with the ground truth box is higher than 0.7 or if it is the highest. If IoU is lower than 0.3, it is classified as background.

$$\text{IoU} = \frac{\text{anchor} \cap \text{ground truth box}}{\text{anchor} \cup \text{ground truth box}}. \quad (1)$$

An RoI (Region of Interest) maxPooling layer is used to fit different size proposed regions that are derived from the RPN to the same size. After the RoI pooling process, the softmax classifier, which classifies the occupants, and the box regressor, which estimates the bounding box, are trained. Therefore, we used the following multitask loss function for training, which is the sum of the $\mathscr{L}_{\text{clf.}}$ (loss function for classification) and the $\mathscr{L}_{\text{bbox}}$ (loss function for bounding box detection).

$$\mathscr{L} = \mathscr{L}_{\text{clf.}} + \mathscr{L}_{\text{bbox}},$$

$$\mathscr{L}(p_i, c_i) = \frac{\sum_i \mathscr{L}_{\text{clf.}}(p_i, p_i^{g.t.})}{N_{\text{clf.}}} + \frac{\lambda \sum_i p_i^{g.t.} \cdot \mathscr{L}_1(c_i - c_i^{g.t.})}{N_{\text{bbox}}}, \quad (2)$$

where p_i is the predicted probability of anchor i, which is an object, and $p_i^{g.t.}$ is the ground truth label of whether anchor i is an object or a background. c_i indicates the predicted four parameterized coordinates of anchor i: x, y position, width, and height. $c_i^{g.t.}$ is the ground truth coordinate of anchor i, and $N_{\text{clf.}}$ and N_{bbox} represent the normalization term, which is set to be the minibatch size and the number of anchor

locations, respectively. λ is the balancing parameter that makes $\mathscr{L}_{\text{clf.}}$ and $\mathscr{L}_{\text{bbox}}$ of approximately the same weight. In case of the bounding box regression, the coordinates and the training through the $\mathscr{L}_1$ loss function are estimated as follows:

$$\begin{aligned} c_x &= \frac{x - x_a}{w_a}, \\ c_y &= \frac{y - y_a}{h_a}, \\ c_w &= \log\frac{w}{w_a}, \\ c_h &= \log\frac{h}{h_a}, \\ c_x^{g.t.} &= \frac{x^{g.t.} - x_a}{w_a}, \\ c_y^{g.t.} &= \frac{y^{g.t.} - y_a}{h_a}, \\ c_w^{g.t.} &= \log\frac{w^{g.t.}}{w_a}, \\ c_h^{g.t.} &= \log\frac{h^{g.t.}}{h_a}, \end{aligned} \tag{3}$$

where x, y, w, and h are the coordinates of the anchor and the bounding box: x, y position, width, and height, respectively. The variables x, x_a, and $x^{g.t.}$ indicate the predicted bounding box, anchor box, and ground truth box, respectively, and their meaning is the same as the variables y, w, and h.

In order to train the effective classifier using the Faster RCNN, selecting a pretrained CNN for image feature extraction is important. In this study, we used the Inception-v3 network, which has high accuracy, small model size, and short calculation time, to derive the feature map of the input image used in the RPN and the occupant classification process [22]. In addition, transfer learning was performed using a pretrained Inception-v3 network of over 1 million images in the ImageNet database. The Inception-v3 network is an improved version of GoogLeNet [23], which was released in 2014 with 23.9 million parameters. GoogLeNet features an inception module that allows dense processing of matrix calculations while reducing the connectivity between the nodes in the network configuration. In addition, Inception-v3 improves the kernel used for convolution operations by introducing a new structured inception module that uses the 5 x 5 convolution operation twice for the 3 x 3 convolution operation and replaces the 3 x 3 convolution operation with the 1 x 3 and 3 x 1 convolution operations to reduce the computational complexity. In addition, convolution operations and pooling processes were performed in parallel, and then in concatenation, to improve the representational bottleneck, which is a phenomenon in which the amount of information is greatly reduced when the dimension is reduced excessively in a neural network. Moreover, according to [24], Inception-v3 achieved an accuracy of over 78.1% on ImageNet data sets. To apply Inception-v3 to the Faster RCNN structure, we removed the last three layers, which perform image classification, from the Inception-v3 network and added a feature extraction layer. Afterward, to form the Faster RCNN, a new classification layer and the RPN were added to fit the occupant label defined in this study. The overall structure of the model that detects occupants from single side images is shown in Figure 4.

4. Results and Discussion

Randomly sampled from 1,246 image sets, 1,000 image sets were used for model training, and 246 image sets were used for the detection accuracy test to analyze the vehicle occupant detection framework's performance. Model training was performed using a Stochastic Gradient Descent with momentum solver with a momentum of 0.9, and the learning rate was fixed at 0.001 for the entire training process. Previously, a 4-step method was used to train the Faster RCNN; the training of this study model was performed using an end-to-end method, which has improved the training efficiency.

In addition, to evaluate the efficiency of the labeling method presented in this study, we compared it to a model that uses human labeling methods, using the same data set and the same network structure. The human labeling method is a technique for labeling each person present in an image as an individual object. This method is generally used in vehicle occupant detection area and human detection tasks [12–17]. Two scenarios were used to compare the detection accuracy between the two labeling methods. The first scenario uses both side cameras, assuming an environment that requires high accuracy. The second scenario only uses one camera, assuming that the installation environment and cost are limited. In general, to use the HOV lane enforcement system, it is possible to simply calculate accuracy as a binary case that determines whether the total number of occupants in a vehicle is more than two or more than three, depending on the HOV lane types. If detailed seat occupant detection is possible, the system use increases. Therefore, in this study, the accuracy of the binary case, as well as the accuracy of the detected number of occupants in both the front and rear seats, was also calculated and compared. In the case of the model using the occupant labeling method proposed in this study, the detection result is derived from the number of occupants in the front and rear seats without additional postprocessing. However, in the case of the model trained by the comparative labeling method, the number of occupants in the front and the rear seats is recalculated using the human detection results. To distinguish between the front and rear seats, the B-pillar position in each image is calculated from the distance between the detection results. All the methods in this study were implemented using MATLAB 2019b and trained and

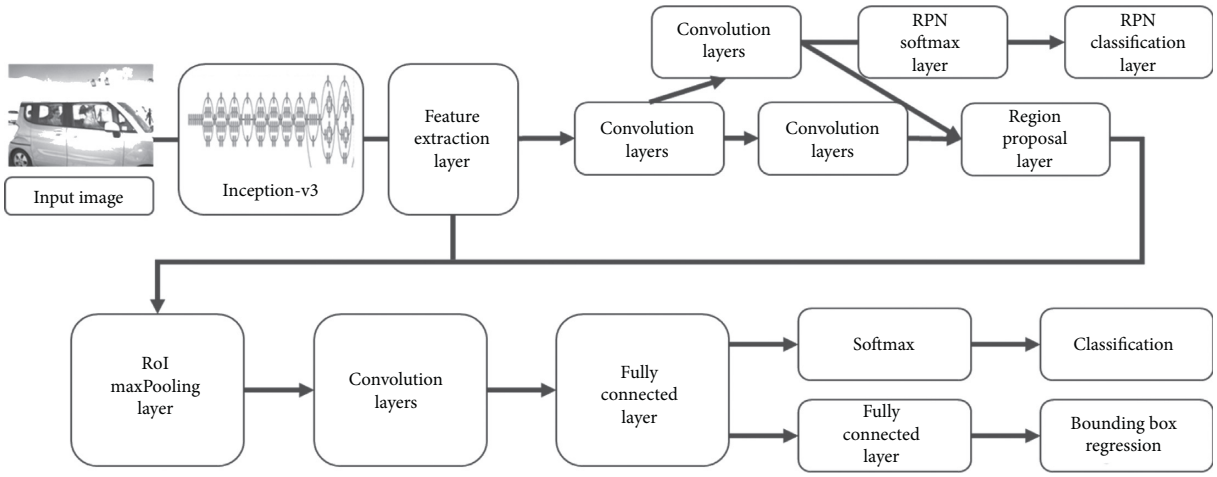


Figure 4: Description of Faster RCNN for one side image.

tested in a Dual Intel® Xeon® Silver 4114 CPU @ 2.20 GHz, 32 GB ram, and single NVIDIA GeForce RTX 2080 Ti computing environment.

The model training time of both labeling methods was about 7 hours for 1 K iterations. When testing these models, the occupant labeling model took an average of 3.4 seconds to output the detection results per an image set; however, it took the human labeling model about 7.6 seconds, more than twice the time of the occupant labeling method. An example of the vehicle occupancy detection test results for each model is shown in Figure 5. The occupant labeling model shows how many people were in the front and rear seats, while the human labeling model shows all the detected people and distinguishes the front seat from the rear one by the virtual B-pillar.

Table 1 compares the results of the occupant detection accuracy of the two models for 246 left and right side image sets. The presented occupant labeling method had a relatively high accuracy in all cases. The human labeling model was also highly accurate in the binary case when detecting two or more persons, but its accuracy was very low when detecting the actual number of occupants. There was an especially big difference in the detection rate of the number of passengers in the rear seat; the proposed labeling method robustly detects the passengers, even when parts of them are hidden in the captured images. In the human labeling method, the neural network learns a person's head and shoulders. When many people occupy a vehicle, especially in the rear seat, some parts of the passengers are often blocked, so it is difficult to identify accurate features. If there are several people riding in a vehicle, the rear seat often covers a part of one or more passengers. Thus, it is difficult to identify accurate human features. The detection accuracy of occupants in the proposed model in this study is 98% for the binary case and 91% for the counting case, which is higher than the accuracy level of the proposed model in [17], which was considered a state-of-the-art occupant detection accuracy.

The confusion matrix allows a more detailed analysis of the detection results of each model. In Figure 6, we present the confusion matrix of the test results for both models. The two matrices on the left are the model results using the occupant labeling method presented in this study, and the two matrices on the right are the model results using the human labeling model. The front and rear seat detection results for each model are shown in two confusion matrices. First, the front seat results are compared with 99.59% and 82.93%, respectively. In the occupant labeling model, one person was incorrectly detected as two people in one instance. However, there were four cases in which the control group detected that two people boarded while one person actually boarded, but 38 cases detected that one person boarded when two people boarded. A person in the passenger seat might be assumed to be a part of the vehicle or hidden by the driver and not be correctly detected as a person. Furthermore, the difference between the rear seat detection accuracy of the two models was 91.06% and 66.26%, respectively, which is greater than the front seat detection accuracy difference. In most cases, the proposed model in this study accurately detects the number of occupants, and the false detection results are maintained at ±1 person in comparison with the actual number. Therefore, it is evident that this model can robustly detect the results for the binary case. On the contrary, in the control model, the detection accuracy was very low when 3 people or more were on board, and there were many results that showed more than 2-person differences from the actual number of passengers. This is similar to the front seat detection result; the occupant labeling method was more effective when learning the appearance of part of the rear seat passengers. Generally, when using human labeling methods, it is difficult to detect people if some parts of them are hidden.

Instead of using both left and right images, the scenario performed detection using only one image on the right side, and the results are presented in Table 2. In the case of detecting occupants using only one camera image, the proposed model showed better results than the human detection method, similar to those in the case of using two camera images. Besides, when using one camera instead of two cameras, the accuracy of the rear seat decreased because the rear seat occupants are often concealed when using

(a) (b)

Figure 5: Examples of vehicle occupancy detection results for both models. (a) Occupant labeling result. (b) Human labeling result.

Table 1: Detection accuracy for both models using two images.

Binary case			
Labeling method	2+		3+
Occupant	**99.2%**		**97.2%**
Human	97.6%		82.1%
Counting case			
Labeling method	Front seat	Rear seat	Total
Occupant	**99.6%**	**91.1%**	**90.7%**
Human	82.9%	66.3%	61.8%

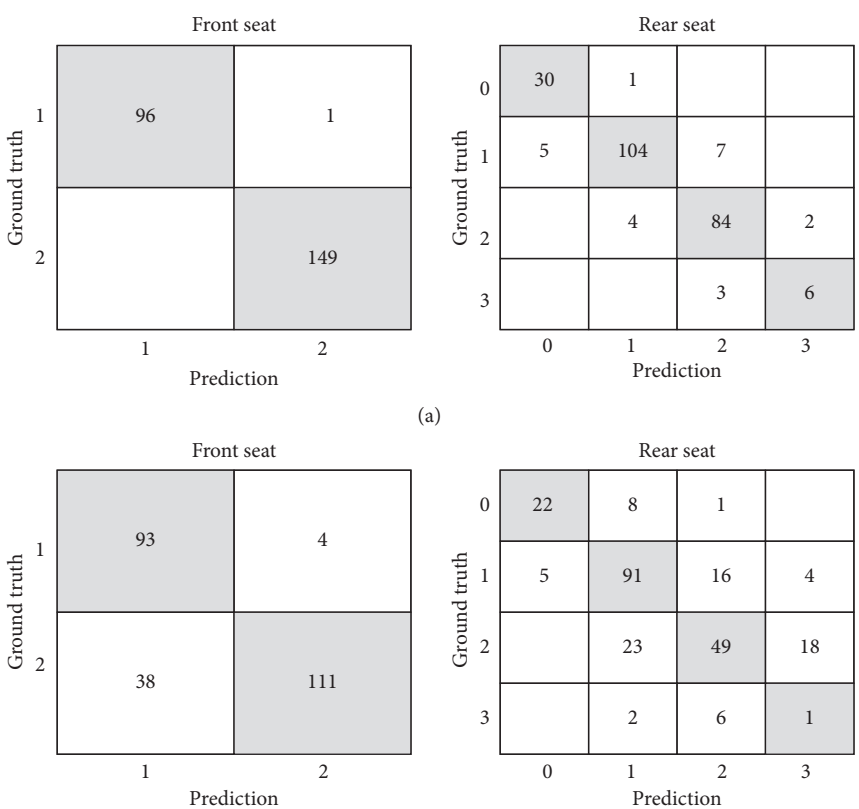


Figure 6: Confusion matrices for tested models. (a) Occupant labeling result. (b) Human labeling result.

Table 2: Detection accuracy for both models using one image

Binary case			
Labeling method	2+		3+
Occupant	**98.8%**		**93.5%**
Human	89.4%		74.8%
Counting case			
Labeling method	Front seat	Rear seat	Total
Occupant	**99.6%**	**87.4%**	**87.0%**
Human	82.9%	54.1%	47.2%

images from only one side, and the image from the opposite side cannot compensate for the smaller number of images. Nevertheless, the single-camera model in this study showed a level of accuracy of 87%, which is similar to that in [17], which showed the highest accuracy (88%) when using two cameras. In particular, in the binary case, the model's accuracy is more than 90%, so a single-camera detection model could be used effectively in an HOV enforcement system. Therefore, according to the purpose and environment of use, it is possible to use the proposed occupancy detection algorithm flexibly in this study.

5. Conclusions

To overcome increasing traffic and encourage carpooling, many governments use HOV lanes and provide discounted toll prices for cars that have multiple passengers. However, such systems usually determine the number of passengers in each vehicle by employing police officers or employees at the roadsides or near the toll booth cashiers. Thus, such human-resource-based occupancy detection systems lead to an operating budget burden and lower accuracy. Due to these limitations, several studies have attempted to achieve automated vehicle occupancy detection systems in a variety of ways, including the use of in-vehicle sensors or out-of-vehicle images. However, the image acquisition difficulty and the weakness of image processing technologies make implementing such detection systems hard to achieve.

To compensate for the shortages of previous research, this study suggests a new labeling method that detects passengers based on the number of occupants in each row of the vehicle instead of using human (or face) and window labeling. This new labeling method achieves Faster RCNN detection in a short time and with high accuracy. Also, this study had two scenarios: (i) using two cameras; (ii) using a one side camera due to the possible difficulties of setting two cameras on each side of the road in some areas. Each scenario has two cases: (i) binary: 1 or 2 and more ('2+')/1 to 2 or 3 and more ('3+'); (ii) counting the actual passenger numbers. Synthetically, the 2+ case had a similar detection accuracy to that of the occupant labeling method (99%), which this study suggests, and to that of the human labeling (97%) method, which is the usual detection method. However, the 3+ case showed a bigger gap (15%) between the two labeling methods, and the counting case had a huge difference between the two methods: occupants (91%) and humans (62%). The counting case is the actual number of passengers and the actual detection accuracy of the automated detection systems. The one side camera scenarios had similar patterns when it came to the detection results, but generally the accuracy was lower than when two cameras were used. In order, 2+, 3+, and the counting case scenarios had bigger differences with the labeling method, the occupant label had a detection accuracy of 87%, and the human labeling method had an accuracy of 46% at the counting case.

Since higher detection accuracy was achieved with the actual system, this study is important for further research on the way to increase the accuracy ratio. In the future, we will try various machine learning methodologies and neural networks to get more advanced results based on the new labeling method.

Data Availability

The data used to support the findings of this study have not been made available because of GnT Solution's policy.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D Program (Project no. 0002246).

References

[1] K. Jang, K. Chung, D. R. Ragland, and C.-Y. Chan, "Safety performance of high-occupancy-vehicle facilities," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2099, no. 1, pp. 132–140, 2009.

[2] L. Markkula, "HOV lanes: issues and options for enforcement," Tech. Rep. FHWA-AZ-04-552, Arizona Department of Transportation, Phoenix, AZ, USA,, 2004.

[3] S. Schijns and P. Mathews, "A breakthrough in automated vehicle occupancy monitoring systems for hov/hot facilities," in *Proceedings of the 12th HOV Systems Conference*, vol. 1, Houston, TX, USA, April 2005.

[4] S. Gautama, S. Lacroix, and M. Devy, "Evaluation of stereo matching algorithms for occupant detection," in *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No.PR00378)*, pp. 177–184, Corfu, Greece, September 1999.

[5] M. Devy, A. Giralt, and A. Marin-Hernandez, "Detection and classification of passenger seat occupancy using stereovision," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, pp. 714–719, Dearborn, MI, USA, October 2000.

[6] M. Klomark, "Occupant detection using computer vision," M. S. thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2000.

[7] Y. Lu, C. Marschner, L. Eisenmann, and S. Sauer, "The new generation of BMW child seat and occupant detection system SBE2," *International Journal of Automotive Technology*, vol. 3, no. 2, pp. 53–56, 2002.

[8] Y. Owechko, N. Srinivasa, S. Medasani, and R. Boscolo, "High performance sensor fusion for vision-based occupant detection," in *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, vol. 2, pp. 1128–1133, Shanghai, China, October 2003.

[9] M. E. Farmer and A. K. Jain, "Occupant classification system for automotive airbag suppression," in *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Madison, WI, USA, June 2003.

[10] F. Erlik Nowruzi, W. A. El Ahmar, R. Laganiere, and A. H. Ghods, "In-vehicle occupancy detection with convolutional networks on thermal images," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, June 2019.

[11] J. W. Wood, G. G. Gimmestad, and D. W. Roberts, "Covert camera for screening of vehicle interiors and HOV enforcement," in *Proceedings of the Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Defense and Law Enforcement II*, pp. 411–420, Orlando, FL, USA, September 2003.

[12] X. Hao, H. Chen, and J. Li, "An automatic vehicle occupant counting algorithm based on face detection," in *Proceedings of the 2006 8th International Conference on Signal Processing*, vol. 3, Beijing, China, November 2006.

[13] B. Xu, P. Paul, Y. Artan, and F. Perronnin, "A machine learning approach to vehicle occupancy detection," in *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, vol. 2014, pp. 1232–1237, Qingdao, China, November 2014.

[14] Y. Artan, P. Paul, F. Perronin, and A. Burry, "Comparison of face detection and image classification for detecting front seat passengers in vehicles," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pp. 1006–1012, Steamboat Springs, CO, USA, March 2014.

[15] Y. Artan, O. Bulan, R. P. Loce, and P. Paul, "Passenger compartment violation detection in HOV/HOT lanes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 395–405, 2016.

[16] B. Balci, B. Alkan, A. Elihos, and Y. Artan, "Front seat child occupancy detection using road surveillance camera images," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1927–1931, Athens, Greece, October 2018.

[17] A. Kumar, A. Gupta, B. Santra et al., "VPDS: an AI-based automated vehicle occupancy and violation detection system," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9498–9503, 2019.

[18] J. Lim, S. Kim, S. Kang, and C. Kim, "Development occupancy detection in moving vehicle using computer vision technology," in *Proceedings of the US-Korea Conference on Science, Technology and Entrepreneurship*, Chicago, IL, USA, August 2019.

[19] A. Sachan, "Zero to hero: guide to object detection using deep learning: faster R-CNN," 2020, https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*vol. 39, no. 6, , pp. 1137–1149, NIPS, 2015.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[22] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2017, http://arxiv.org/abs/1605.07678.

[23] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Boston, MA, USA, June 2015.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8897700, 15 pages
https://doi.org/10.1155/2020/8897700

Research Article

Risk Prediction for Ship Encounter Situation Awareness Using Long Short-Term Memory Based Deep Learning on Intership Behaviors

Jie Ma,[1,2,3] Wenkai Li,[1] Chengfeng Jia,[1] Chunwei Zhang,[1] and Yu Zhang[4]

[1]*School of Navigation, Wuhan University of Technology, Wuhan 430063, China*
[2]*Hubei Inland Shipping Technology Key Laboratory, Wuhan 430063, China*
[3]*National Engineering Research Center for Water Transportation Safety, Wuhan 430063, China*
[4]*School of Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China*

Correspondence should be addressed to Chengfeng Jia; jcf@whut.edu.cn

Received 3 September 2020; Revised 20 October 2020; Accepted 9 November 2020; Published 29 November 2020

Academic Editor: Petr Dolezel

Encounter risk prediction is critical for safe ship navigation, especially in congested waters, where ships sail very near to each other during various encounter situations. Prior studies on the risk of ship collisions were unable to address the uncertainty of the encounter process when ignoring the complex motions constituting the dynamic ship encounter behavior, which may seriously affect the risk prediction performance. To fill this gap, a novel AIS data-driven approach is proposed for ship encounter risk prediction by modeling intership behavior patterns. In particular, multidimensional features of intership behaviors are extracted from the AIS trace data to capture spatial dependencies between encountering ships. Then, the challenging task of risk prediction is to discover the complex and uncertain relationship between intership behaviors and future collision risk. To address this issue, we propose a deep learning framework. To represent the temporal dynamics of the encounter process, we use the sliding window technique to generate the sequences of behavioral features. The collision risk level at a future time is taken as the class label of the sequence. Then, the long short-term memory network, which has a strong ability to model temporal dependency and complex patterns, is extended to establish the relationship. The benefit of our approach is that it transforms the complex problem for risk prediction into a time series classification task, which makes collision risk prediction reliable and easier to implement. Experiments were conducted on a set of naturalistic data from various encounter scenarios in the South Channel of the Yangtze River Estuary. The results show that the proposed data-driven approach can predict future collision risk with high accuracy and efficiency. The approach is expected to be applied for the early prediction of encountering ships and as decision support to improve navigation safety.

1. Introduction

Water traffic has become increasingly busy with the rapid development of the shipping industry in recent years, which has led to an increased risk to individuals and society in terms of various aspects, especially ship-ship collision accidents. Owing to the frequent occurrences and serious consequences of collisions, research on reducing collision accidents from both theoretical and practical points of view has always been a major topic of concern for navigational experts and scholars. Perceiving risk and predicting encounter situations between ships are crucial for the prevention of collision accidents, especially in busy traffic areas, where congested ships sail relatively close to each other [1].

To understand the risk level and take actions to decrease the possibility of collisions occurring in the waters, numerous efforts have been devoted to the risk analysis and assessment of ship collisions. Some focus on risk surveys among maritime experts and the conduct of qualitative collision risk analyses, primarily through empirical studies. Cohen et al. [2] used highly stressful training scenarios generated by a ship simulator to measure the heart rates of participants to estimate the collision risk. Chin and Debnath

[3] examined the risks of different ship types by developing a survey conducted by Singapore port pilots under both day and night conditions. However, the above qualitative methods do not take into account the ship navigation data, so it is difficult for them to reflect highly dynamic and continuous vessel movement as well as the evolutionary collision risk trends.

In recent years, with the wide application of automatic identification systems (AIS) in water traffic control and surveillance, AIS data have been proven to be a valuable source of ship behavior monitoring and analysis [4–6]. The AIS can transmit motion information (e.g., speed, course, etc.) between ships, from ships to shore, or vice versa. This makes it possible to quantitatively analyze collision risk by means of massive AIS data. Silveira et al. [7] used AIS data to model traffic patterns off the coast of Portugal, based on which the probability of a ship collision occurring was calculated. A related method was adopted by Christian and Kang [8] to develop a probabilistic risk assessment. In addition, the motion data obtained from AIS can be used to calculate the distance to the closest point of approach (DCPA) and time to closest point of approach (TCPA), which can quantify the collision risk from spatial and temporal aspects, respectively. Ahn et al. [9] defined the membership functions of DCPA and TCPA based on the simulation results. Collision-avoidance maneuvers were then obtained using multilayer perceptron neural networks. Similarly, Hwang et al. [10] designed a fuzzy collision-avoidance expert system, where the DCPA and TCPA were considered simultaneously. With the help of their system, ships can be advised to make proper maneuvers to avoid collisions at the right time.

However, as navigational experts and some studies have found, the DCPA and TCPA do not fully reflect the actual collision risk level, and using only these two parameters may lead to misjudgments regarding the collision risk [11, 12]. Therefore, modeling the collision risk using multiple parameters has gradually been adopted by most researchers. Ren et al. [13] presented a linear model for evaluating ship collisions, which considered several factors, such as the ship type, velocity, and route. Silveira et al. [14] estimated the distances between ships by using sampled positions, courses, and speeds, based on which the number of varying collision candidates was evaluated by comparing it with a predefined collision diameter. Zhang et al. [15] developed a vessel conflict ranking operator (VCRO) model, which considered the relative ship speeds, the course difference, and distance between two ships. Then, the Northern Baltic Sea AIS data were used to assess the risk of a near-miss collision, and the results indicate that the model is adequate for ranking the encounters. Based on the VCRO model, Zhang et al. [16] combined the density complexity of open waters with the multivessel VCRO model to assess the regional near-miss collision risk.

The ship domain is supposed to be a feasible metric to make collision risk predictions based on the assumption that the risk of collision is high when the ship's domain is invaded by the target ship. Szlapczynski [17] proposed a novel method to measure collision risk by adopting the concept of an ellipse-shaped ship domain. Further, Szlapczynski and Szlapczynska [18] addressed the domain violation problem by combining two parameters, that is, the degree and time of domain violation, to offer an intuitive assessment of the collision risk. Wu et al. [19] also employed the ship domain violation rule to study the frequency of ship conflicts, which considered elliptical and circular domains individually, and a series of hot spots with high collision risk in the Sabine-Neches Waterway were identified by using the two domain types. Wang [20] proposed a novel ship domain model termed the fuzzy quaternion ship domain (FQSD). The domain sizes are determined by the quaternion, including the forward, aft, starboard side, and port side radius. The FQSD model uses fuzzy boundaries (e.g., the ship boundary could be linear or nonlinear as well as thin or fat) to estimate the collision risk, aiming at providing a reasonable and dependable evaluation method. By taking advantage of the FQSD model, Qu et al. [21] estimated the number of ship domain overlaps to evaluate the collision risk in the Singapore Strait, assuming that increased ship domain overlap indicates a higher ship collision probability. However, the ship domain uncertainty will seriously affect prediction performance. Several measures, such as the length and width of the encounter ship, which should be known when calculating the ship domain are not always available. In addition, most collision risk prediction approaches based on ship domains assume that the speed and course of the ship are constant at the moment of sampling [22], which does not sufficiently take into account the evolutionary factors of an encounter process affecting the risk.

The primary limitation of prior studies on the risk of ship collisions is that they cannot address the uncertainty of the encounter process when neglecting the complex motions constituting the dynamic behavior of encountering ships. Thus, it is necessary to incorporate the spatiotemporal behaviors of ships encountering each other (intership behavior) to make the risk prediction more reasonable because the intership behavior will determine the subsequent risk state to a certain extent with the evolutionary process of the ship encounter but was rarely considered and implemented in previous research. To bridge this gap, we propose a novel AIS data-driven approach for ship encounter risk prediction by modeling intership behavior patterns. The primary contributions of this study are summarized as follows:

(i) Intership behavior is essentially a stochastic process consisting of the motion behaviors of any encountered ships. Following this rationale, this paper proposes modeling intership behavior by transforming the AIS traces into a sequence of behavioral features by combining a fixed set of parameters, including the relative velocity, course difference, and relative distance as well as three azimuthal types. With this time series structure, the process of ships encountering each other and the corresponding spatiotemporal dynamics can be effectively characterized.

(ii) As previously discussed, ship collisions are often closely related to the navigator's behavior. Thus, a

novel method to model the relationship between intership behavior and collision risk involvement is necessary to accurately predict the risk. To address this challenge, we relate the sequence of behavioral features involving a specified time window to the future risk level. Then, the mapping between them can be established through a supervised learning approach, and the problem of risk prediction is formed as a time series classification task [23], which makes the prediction process easier to implement by taking full advantage of the benefits of data-driven modeling with AIS. Inspired by the recent achievements of long short-term memory (LSTM) networks for various time series learning tasks such as text categorization [24, 25] and trajectory prediction [26, 27], we extend them to our mapping modeling between the intership behavior and the collision risk. To the best of our knowledge, we are the first to address this issue through the utilization of LSTM networks.

(iii) With our proposed approach, the potential collision risk associated with the uncertain encounter process could be recognized and identified at an early stage. Thus, early warnings can be provided so that ship officers have sufficient time to react to emergencies and take evasive actions in advance. Additionally, the outcome of this research can provide useful support to human operators in charge of large and crowded water areas and encourage safe navigation under specific scenarios to reduce the incidence of ship collisions.

The remainder of this paper is organized as follows: First, we provide a brief description of the issue of risk prediction in Section 2. Next, Section 3 develops a thorough discussion regarding the extraction of AIS data as well as constructing the sequence of behavioral features. The ship encounter risk prediction frameworks are proposed in Section 4. Finally, Section 5 is dedicated to a summary of our numerical results and a discussion of the model's performance.

2. Problem Formulation

The goal of designing the methodological framework is to investigate the key issues (e.g., ship encounter risk prediction) affecting the intership behavior. The collision risk level of the encountering ships at time t is represented by R_t. R_t is divided into five categories according to the risk level from low to high, and class labels of 1, 2, 3, 4, and 5 represent the following:

$$\mathrm{CRI} = \begin{cases} 1, & \text{low risk level,} \\ 2, & \text{low} - \text{middle risk level,} \\ 3, & \text{middle risk level,} \\ 4, & \text{middle} - \text{high risk level,} \\ 5, & \text{high risk level.} \end{cases} \tag{1}$$

We denote these risk levels as follows:

(i) Low risk level: A situation where risk begins to be present and two ships are free to maneuver.

(ii) Low-middle risk level: A situation in which the ships approaching each other have a collision risk and the given-way ship should maneuver in advance.

(iii) Middle risk level: A situation in which a safe passing distance cannot be ensured if only the given-way ship fully maneuvers.

(iv) Middle-high risk level: A situation in which collision cannot be avoided if only the given-way ship fully maneuvers.

(v) High risk level: A situation in which two ships should fully maneuver to avoid the collision.

In this study, the collision risk can be defined as a continuum spectrum of colors, as shown in Figure 1. This spectrum ranges from the safest situation (a near-zero chance of collision) to the riskiest situation during encounters (both ships need to take evasive actions to avoid collision). A collision risk index (CRI) [28] was employed to calculate the risk spectrum. In terms of collision avoidance, the CRI is essential for a ship officer to evaluate the risk of a ship encounter as well as for performing an evasion strategy [29].

As previously mentioned, the collision risk could be affected by the uncertain and complex behavior of encountering ships. A ship encounter is essentially a dynamic evolutionary process commonly utilized to perceive the situation of encountering ships. The evolution of the encounter process is subjected to the specific motions of each ship as well as pairwise exchanges of influences between the ships, thus indicating that the spatiotemporal kinematics of the ships involved in the encounter have dependency and correlation. To associate the collision risk prediction with the evolution of the encounter process, we aim to model the relationship between the sequence of behavioral features and the future risk level. For one encounter pair, we denote the behavioral features as follows:

$$\begin{aligned} U &= \left[u^1, \ldots, u^i, \ldots, u^N\right], \\ u^i &= \left[V_R^i, A^i, D_{ot}^i, \alpha^i, \alpha_o^i, \alpha_t^i\right]^T, \end{aligned} \tag{2}$$

where U is an $N \times 6$-dimensional variable composed of u^i, where N represents N sampling points. V_R^i, A^i, D_{ot}^i are relative velocity, course difference, and relative distance between two ships, respectively. $\alpha^i, \alpha_o^i, \alpha_t^i$ are three types of azimuths (the details are introduced in Section 3). If the time window is 2δ and the sliding step is δ, then the entire track U can be divided into L time windows. Therefore, an encounter process consisting of a sequence of behavioral features can be reformulated as follows:

$$w^t = \left[u^{t-2\delta+1}, u^{t-2\delta+2}, \ldots, u^t\right], \tag{3}$$

where w^t represents the observation window with length of 2δ before time t. The goal of this study is to predict the risk level of a ship at a future time ΔT, so we need to match w^t with $R^{t+\Delta T}$ and generate the pairs of sample datasets

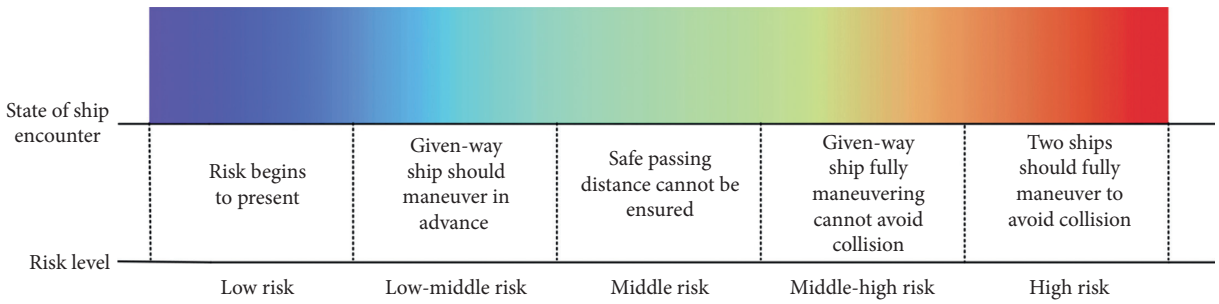


Figure 1: Spectrum of the collision risk level.

$(w^t, R^{t+\Delta T})$. We want to find a function f that can best model the relationship between w^t and $R^{t+\Delta T}$:

$$f\colon\ w^t \longrightarrow R^{t+\Delta T}. \tag{4}$$

By means of equation (5), the issue of risk prediction can be transformed into a time series classification task. To evaluate the model prediction, a confusion matrix was designed to assess the predictability. The size of the square matrix represents the categories of various risk levels. Table 1 shows the confusion matrix with five risk levels. As presented in Table 1, each diagonal element of the confusion matrix represents the correct category; for example, T_L is the proportion of low risk level that is correctly predicted.$\mathrm{FM_L}$ represents False Middle give Low, which is the proportion of low risk level that is wrongly predicted as middle risk level. In addition, the misclassification error rate (MER) is employed to estimate the overall performance of the model. The MER can be obtained by comparing the predicted risk level with the actual risk level as follows:

$$\mathrm{MER} = \frac{1}{N_R}\sum_{1}^{N_R}\left(\hat{R}_s \neq R_s\right), \tag{5}$$

where N_R is the number of the windows. Furthermore, a tenfold cross-validation method is utilized to obtain the best model, which has been empirically shown to yield estimates that suffer neither from overly high bias nor from excessively high variance [30].

Table 1: Confusion matrix for prediction result of risk level.

Risk level	L	LM	M	MH	H
L	$\mathrm{T_L}$	$\mathrm{FL_{LM}}$	$\mathrm{FL_M}$	$\mathrm{FL_{MH}}$	$\mathrm{FL_H}$
LM	$\mathrm{FLM_L}$	$\mathrm{T_{LM}}$	$\mathrm{FLM_M}$	$\mathrm{FLM_{MH}}$	$\mathrm{FLM_H}$
M	$\mathrm{FM_L}$	$\mathrm{FM_{LM}}$	$\mathrm{T_M}$	$\mathrm{FM_{MH}}$	$\mathrm{FM_H}$
MH	$\mathrm{FMH_L}$	$\mathrm{FMH_{LM}}$	$\mathrm{FMH_M}$	$\mathrm{T_{MH}}$	$\mathrm{FMH_H}$
H	$\mathrm{FH_L}$	$\mathrm{FH_{LM}}$	$\mathrm{FH_M}$	$\mathrm{FH_{MH}}$	$\mathrm{T_H}$

3. Data Preparation and Feature Extraction

In this section, we describe the process of extracting the behavior features from the original AIS trace data, which can effectively characterize the navigation activities and corresponding spatiotemporal dynamics. The process comprises two components. First, we clean and integrate the enormous volume of original AIS data, by which the AIS data will be purified and selected into a time series structure. Then, through the space-time registration, the synchronous pairwise trajectory of encountering ships can be obtained. Next, we transform the pairwise trajectory data into a sequence of behavioral features by combining a fixed set of parameters.

3.1. Data Preparation. The AIS is an automatic tracking system to improve navigation safety and avoid collision accidents by providing the navigation information of various ships. In general, this navigation information in AIS messages is broadly classified as either dynamic information or static information. The dynamic information includes the ship location (longitude and latitude), speed over ground (SOG), course over ground (COG), destination, and estimated arrival time. The static information contains the ship name, ship maritime mobile service identity (MMSI), ship type, ship size, current time, and other information. As the AIS data contains the above information, it can serve as the data source for understanding the traffic situations [31]. In particular, SOG and COG have substantial impacts on dangerous encounter situations. Many existing studies take SOG and COG into consideration in ship collision risk assessment [32–34]. However, there are some errors in the AIS data, such as messy codes and data irrationalities, which may contribute to misjudgments of collision accidents. Therefore, certain preprocessing methods are essential to ensure the reliability and applicability of the AIS data to gain a better investigation of the collision risk.

3.1.1. Data Cleaning and Trajectory Interpolation. This part aims to eliminate the above-mentioned errors in the AIS data. A mathematical data cleaning method is used. We denote a trajectory Traj as follows:

$$\begin{aligned}\mathrm{Traj} &= \left[\mathrm{traj}^1, \ldots, \mathrm{traj}^m, \ldots, \mathrm{traj}^M\right],\\ \mathrm{traj}^m &= \left[\mathrm{lng}^m, \mathrm{lat}^m, \mathrm{sog}^m, \mathrm{cog}^m\right]^T,\end{aligned} \tag{6}$$

where M is the number of trajectory sampling points. traj^m denotes the four-dimensional vector of the m-th sampling points, which contains the location information and kinematics information of the ship. With this background, the method filters out the outliers by taking the statistics of data distribution statistics into account. Assuming that these parameters are normally distributed, the distribution can be identified by the mean and the variance calculated from the

samples. According to the 3σ rule, the outlier points in the data can be eliminated. Taking the longitude lng as an example, formulas (7)–(9) show how to eliminate outlier points. If equation (10) is satisfied, lng^m of the sampling point traj^m is considered as an abnormal value. lng^m needs to be removed and replaced with blank placeholders.

$$\overline{\text{lng}} = \frac{1}{M}\sum_{1}^{M}\text{lng}^m, \tag{7}$$

$$\varepsilon_{\text{lng}}^m = \text{lng}^m - \overline{\text{lng}}, \tag{8}$$

$$\sigma_{\overline{\text{lng}}} = \sqrt{\frac{1}{M-1}\sum_{1}^{M}\varepsilon_{\text{lng}}^m}, \tag{9}$$

$$\left|\varepsilon_{\text{traj}}^m\right| > 3\sigma_{\overline{\text{traj}}}. \tag{10}$$

Because of the AIS system broadcasting frequency and the above outlier elimination process, there will be some missing data at different time points. That is, the time intervals between the sampling points in a track Traj may not be equal. For example, the time interval between point traj^i and point traj^{i+1} may not be the same as that between point traj^{i+1} and point traj^{i+2}. The purpose of this portion is to form a continuous time series with equal frequencies using the interpolation method. In particular, different interpolation methods are used to fill in the blanks according to the variable sparsity of a track Traj. Through the initial window length of 240 s, the whole track Traj can be divided into L windows to identify its sparsity. The smaller the size of L is, the sparser the data is (i.e., the smaller the sampling frequency is).

(1) If $L < 15$, then the trajectory data is too sparse, and it is difficult to restore the missing information even through the interpolation method. For such cases, we discarded these trajectories.

(2) If $15 \le L < 20$, then the trajectory data are sparse for a portion of the time windows. Thus, we reduce the window length to 120 s to guarantee the density of the data in shorter windows. Then the linear interpolation is selected for the sequences in each shorter window.

(3) If $20 \le L$, it means that the sampling frequency of the data is relatively consistent. For such dense data, the Hermitian cubic interpolation achieves better results than linear interpolation.

As the proposed method is employed to interpolate various sparsity situations of the trajectory data, a continuous time series with equal frequency can be obtained, in which the frequency is 1 Hz.

3.1.2. Pairwise Trajectory Selection. Through data cleaning and trajectory interpolation, a dataset of a fine single trajectory was obtained. To predict the collision risk in an encounter situation, it is necessary to match the pairwise trajectories of these ship pairs. The matching rule takes both time and space constraints into account. Specifically, as shown in Figure 2, these selected pairwise trajectories $(\text{Traj}_a, \text{Traj}_b)$ should have intersections in the time dimension and be close to each other in the space dimension.

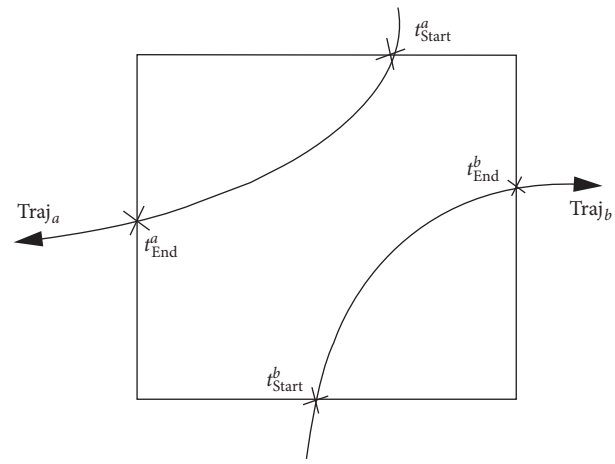


Figure 2: Diagram of space-time constraints during the encounter process.

(1) If $[t^a_{\text{Start}}, t^a_{\text{End}}]$ and $[t^b_{\text{Start}}, t^b_{\text{End}}]$ denote the time intervals of the ship a and ship b respectively, then $[t^a_{\text{Start}}, t^a_{\text{End}}] \cap [t^b_{\text{Start}}, t^b_{\text{End}}] \neq \varnothing$.

(2) If $D_{ab} = [d^1, d^2, \ldots, d^t]$ represents the relative distance between two ships, $d_{\max}$ is the distance threshold for assessing the encounter between ships. Then, $\forall d^t \in D_{ab}, d^t < d_{\max}$.

Those trajectories that are subject to the above two constraints can be selected as pairwise samples. In addition, according to the experience of experts and the definition of an encounter, $d_{\max} = 6$.

In crossing situation, two ships are crossing to involve a collision risk. One ship is coming from either the left or right direction of the other ship's bow, and the relative azimuth between the two ships is 5.7° to 112.5°.

(i) Head-on situation: Two ships are meeting on reciprocal or nearly reciprocal courses to involve a collision risk. One ship sees the other ahead or nearly ahead, and the relative azimuth between two ships is −5.7° to 5.7°.

(ii) Overtaking situation: Two ships are sailing on identical or nearly identical course to involve a collision risk. One ship comes up to another ship from 112.5° to 247.5°.

3.2. Feature Extraction of Intership Behaviors. This part aims to obtain insights into the dynamic encounter process through utilizing a sequence of behavioral features. These features have been established by merging the six parameters in a fixed time window, including the relative velocity, course difference, and relative distance as well as three types of azimuths. The coordinate system presented in Figure 3 is

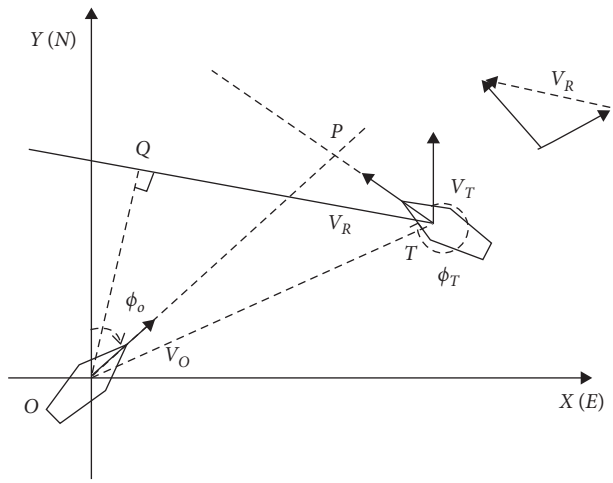


Figure 3: Rectangular coordinate system of own ship and target ships.

established to offer insights into calculating a set of parameters by modeling the spatial relationship between ships encountering each other. As shown in Figure 3, the point O indicates the position of the own ship, and lng_o, lat_o, V_O, and ϕ_O are the longitude, latitude, SOG, and COG of own ship. Moreover, the point T represents the location of the target ship, and lng_t, lat_t, V_T, and ϕ_T are longitude, latitude, SOG, and COG of the target ship. The relative velocity, which is denoted as V_R, is as follows:

$$V_R = |V_O - V_T|. \tag{11}$$

A represents the course difference between the own ship and target ship through incorporating ϕ_O and ϕ_T as follows:

$$A = \begin{cases} \phi_O - \phi_T, & |\phi_O - \phi_T| \le 180^\circ, \\ 360^\circ - (\phi_O - \phi_T), & |\phi_O - \phi_T| > 180^\circ. \end{cases} \tag{12}$$

D_{ot} denotes the relative distance between the two ships, which is estimated by merging with a set of parameters including $\mathrm{lng}_o\mathrm{lat}_o$, lng_t, and lat_t and the Earth radius denoted as R, as follows:

$$\begin{aligned} D_{ot} = {} & R \times \arccos(\sin(\mathrm{lat}_o)\sin(\mathrm{lat}_t)) \\ & + \cos(\mathrm{lat}_o)\cos(\mathrm{lat}_t)\cos(\mathrm{lng}_t - \mathrm{lng}_o), \end{aligned} \tag{13}$$

and α is the true azimuth between the two ships, which can be computed as follows:

$$\alpha = \begin{cases} \arccos\left(\dfrac{\mathrm{lat}_o - \mathrm{lat}_t}{D_{ot}} \times 60\right), & \mathrm{lat}_o > \mathrm{lat}_t, \\ 360^\circ - \arccos\left(\dfrac{\mathrm{lat}_o - \mathrm{lat}_t}{D_{ot}} \times 60\right), & \mathrm{lat}_o \le \mathrm{lat}_t. \end{cases} \tag{14}$$

α_o and α_t denote relative azimuths of two ships, respectively, which are defined as follows:

$$\begin{aligned} \alpha_o &= \alpha - \phi_O, \\ \alpha_t &= \alpha - \phi_T. \end{aligned} \tag{15}$$

Thus, the parameters $w = (V_R, A, D_{ot}, \alpha, \alpha_o, \alpha_t)$ are regarded as the behavioral features of encountering ship pairs.

4. Collision Risk Prediction Model

In this section, we propose a novel collision risk prediction algorithm, which can perceive the potential risk at an early stage by mapping current behavior to future collision risk. To this end, first, the risk level of the current encounter situation is calibrated through the widely used CRI method. Then a deep recurrent neural network structure is used to establish the mapping between the ship's current behavior and future collision risk; then the problem of risk prediction is formed as a time series classification task.

4.1. Collision Risk Calibration. Collision risk calibration is a process used to calculate the risk level for encountering ship pairs. It should be noted that the risk level obtained from the calibration is only an assessment based on the current situation. However, the purpose of this study is to predict future collision risk. Therefore, it is necessary to establish the mapping relationship between the current behavior w^t and the risk level $R^{t+\Delta T}$ after a period of time. ΔT is the prediction horizon, which represents the time interval between observed behavior and predicted risk.

During the training process, a large set of w^t and $R^{t+\Delta T}$ will be prepared to train the model. In this section, the calibration process of the risk level R will be described. As a widely used way of risk calibration, the CRI is used to warn of the collision risk by setting off a collision alarm based on diverse factors influencing the collision risk. In particular, various parameters are taken into account in our calibration process, including the DCPA, TCPA, relative distance, and course difference between two ships [35].

$$\begin{aligned} \mathrm{DCPA} &= D_{ot} \times \sin(\angle \mathrm{OTQ}), \\ \mathrm{TCPA} &= D_{ot} \times \frac{\cos(\angle \mathrm{OTQ})}{V_R}, \\ \mathrm{CRI}_{\mathrm{basic}} &= \left[a_{\mathrm{dcpa}}\left(\frac{\mathrm{DCPA}}{D_s}\right)^2 + a_{\mathrm{tcpa}}\left(\frac{\mathrm{TCPA}}{T_s}\right)^2 + a_d\left(\frac{D_{ot}}{D_s}\right)^2\right]^{-(1/2)}, \\ \mathrm{CRI} &= \mathrm{CRI}_{\mathrm{basic}}\mathrm{F}_{\mathrm{DCPA}}\mathrm{F}_{\mathrm{TCPA}}\mathrm{F}_{\mathrm{cd}}, \end{aligned} \tag{16}$$

where D_{ot} denotes the relative distance between the ships encountering each other. D_s and T_s are the minimum safe distance and time necessary to perform evasive maneuvers; we set them as 0.5 miles and 10 minutes, respectively. Moreover, a_{dcpa}, a_{tcpa}, anda_d are the weights coefficients depending on the state of visibility at sea, the length and beam of the ship, and the type of water area. According to [11], F_{cd} is a multiplier reflecting the encounter danger degree in different encounter situations. Specifically, regarding the course difference between the ships involved in the encounter, the encounter situations can be divided into three categories and the corresponding value of each

multiplier F_{cd} is obtained in Table 2. Moreover, F_{DCPA} and F_{TCPA} are the amplification coefficients of DCPA and TCPA, which are somewhat inversely proportional to the values of DCPA and TCPA. The formulas for calculating the amplification coefficients are given as follows:

$$F_{TCPA} = \exp^{-(TCPA/10)},$$
$$F_{DCPA} = \exp^{-DCPA}. \quad (17)$$

Obviously, from the above equation, CRI is a continuous value. However, continuous CRI values do not directly indicate the urgency of a ship collision risk. In other words, even if we know the value of the CRI, we cannot be certain about the danger level it represents. Here, we apply the different risk stages of ship encounters to divide the CRI into five different risk levels: low (L), low-middle (LM), middle (ML), middle-high (MH), and high (H):

$$\begin{cases} 1, & \text{if } 0 \le \text{CRI} < \tau_1 \text{ [low risk level]}, \\ 2, & \text{if } \tau_1 \le \text{CRI} < \tau_2 \text{ [low} - \text{middle risk level]}, \\ 3, & \text{if } \tau_2 \le \text{CRI} < \tau_3 \text{ [middle risk level]}, \\ 4, & \text{if } \tau_3 \le \text{CRI} < \tau_4 \text{ [middle} - \text{high risk level]}, \\ 5, & \text{if } \tau_4 \le \text{CRI [high risk level]}, \end{cases} \quad (18)$$

where τ_1, τ_2, τ_3, and τ_4 are threshold values that need to be determined to separate different risk levels. We can determine them through analyzing the distribution of CRI, which is computed from AIS data of the encounter ships involved in the encounter. In addition, it has been put forward that the statistical probability of the CRI is equal in each encounter stage [36]. In following this reasoning, we calculate the corresponding CRI values for all the samples by using equation (18). All the CRI values are sorted and divided into five equal intervals according to the frequency. The endpoint of the i-th interval is the threshold τ_i. Figure 4 shows the thresholds selected for each risk level, the left side of Figure 4 illustrates the cumulative probability of the CRI in these time windows, and the right side of Figure 4 counts the number of each encounter stage, which is closely related to the corresponding risk levels. Thereby, the threshold values of the five risk levels are provided as follows:

(1) The CRI values between 0.00 and 0.13 are ranked as the low risk level.

(2) The CRI values between 0.13 and 0.20 are ranked as the low-middle risk level.

(3) The CRI values between 0.20 and 0.28 are ranked as the middle risk level.

(4) The CRI values between 0.28 and 0.45 are ranked as the middle-high risk level.

(5) The CRI values larger than 0.45 are ranked as the high risk level.

Following the above process of risk discretization, the risk level R^t at a different time t is obtained. As mentioned earlier, we will match the behavior sequence w^t with the risk $R^{t+\Delta T}$ to obtain the training set. From the perspective of machine learning classification, w^t is the temporal feature, and $R^{t+\Delta T}$ is the label. Thus, the problem of risk prediction is transformed into a problem of sequence classification. The following section will introduce the sequence classification method used in this paper.

Table 2: The relationship between F_{cd} and course difference.

Course difference	0° – 60°	60° – 150°	150° – 180°
F_{cd}	1	8.5	2.34

4.2. Risk Prediction Model. With the fast development of deep learning, recurrent neural networks (RNNs) have gained great success in recent years [37] in terms of sequence classification. While an RNN has the ability to make full use of the information of the historical input, it is difficult to manage the long-term dependence caused by the fast failure of nodes. As one of the advanced RNNs, LSTM networks address this issue by modifying the internal RNN cell structure. In particular, LSTM contains a set of memory blocks consisting of one or more autocorrelative memory cells and three gates, that is, input, output, and forget gates. In following this structure, a memory block can retain the relevant historical information [38]. Besides, the sequence of behavioral features is considered to be a typical time series; thus, it follows that the issue of risk prediction can be treated as a time series classification task. In view of the above, it is reasonable to think that LSTM networks can provide valuable insight for predicting the collision risk between ships encountering each other. With this modeling framework, an understanding of the sequence of behavioral features and their relationships with collision risk can be achieved.

In our implementation, we assume that w^t is the sequence of behavioral features in the t-th time window; in addition, $R^{t+\Delta T}$ represents the risk level profiles in the $t + \Delta T$-th time window computed in terms of equation (18). With the evolution of the encounter process, the LSTM networks are employed to learn the mapping between w^t and $R^{t+\Delta T}$. Figure 5 shows the modeling framework of this mapping; it can be clearly observed from Figure 5 that the sequence of behavioral features involved in the time window is effectively related to the risk level. Thus, the ship encounter risk prediction is achieved by utilizing the encounter dataset under three encounter situations.

5. Experimental Results and Discussion

5.1. Study Areas. The South Channel intersection waterway, an important and busy shipping channel located on the Yangtze Estuary, was selected as the study area. Figure 6 shows an electronic chart of the South Channel intersection waterway. Figure 6 indicates that a large number of ships in this waterway lead to complex encounter situations. In such a water area with dense traffic flow, the early identification of risk is very important for navigation safety. In this study, we use an AIS dataset for 1729 ships in the South Channel intersection waterway from 07/01/2019 to 08/31/2019. Subsequently, the sequence of behavioral features is

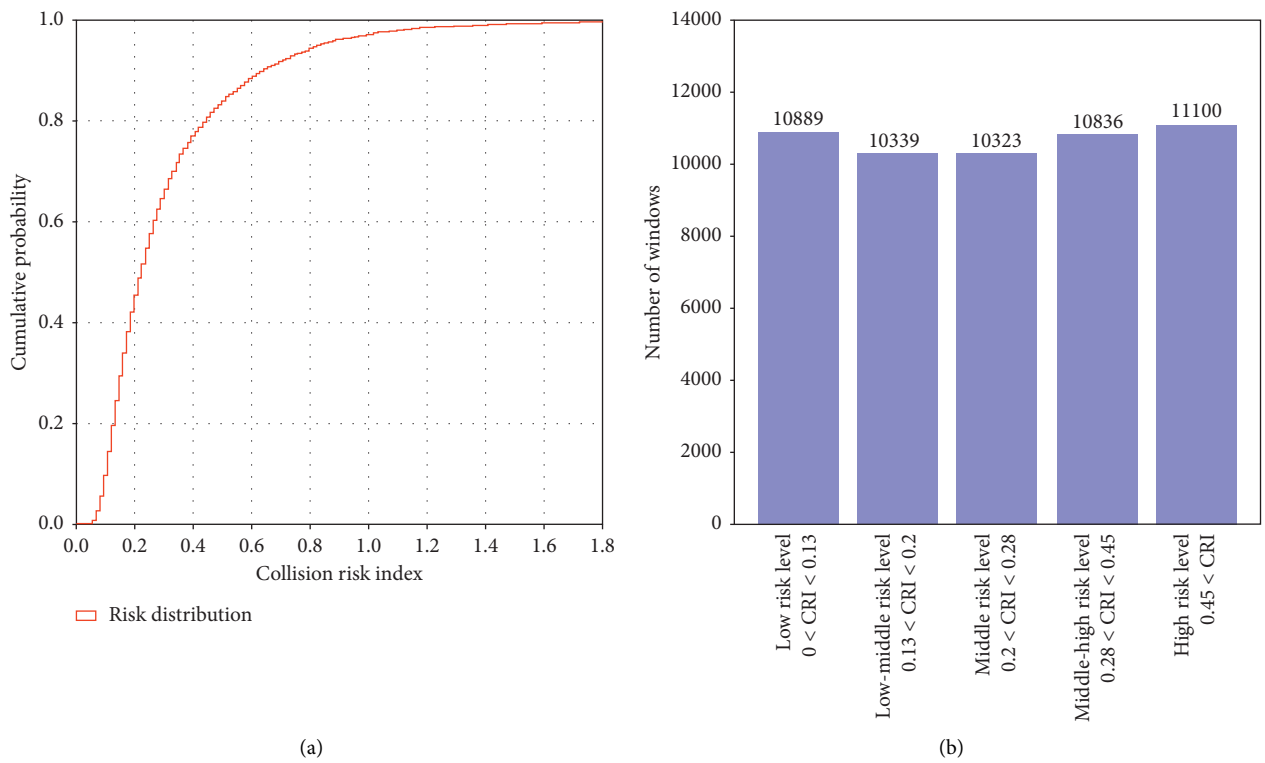


Figure 4: Cumulative probability of CRI and number of CRI in each stage.

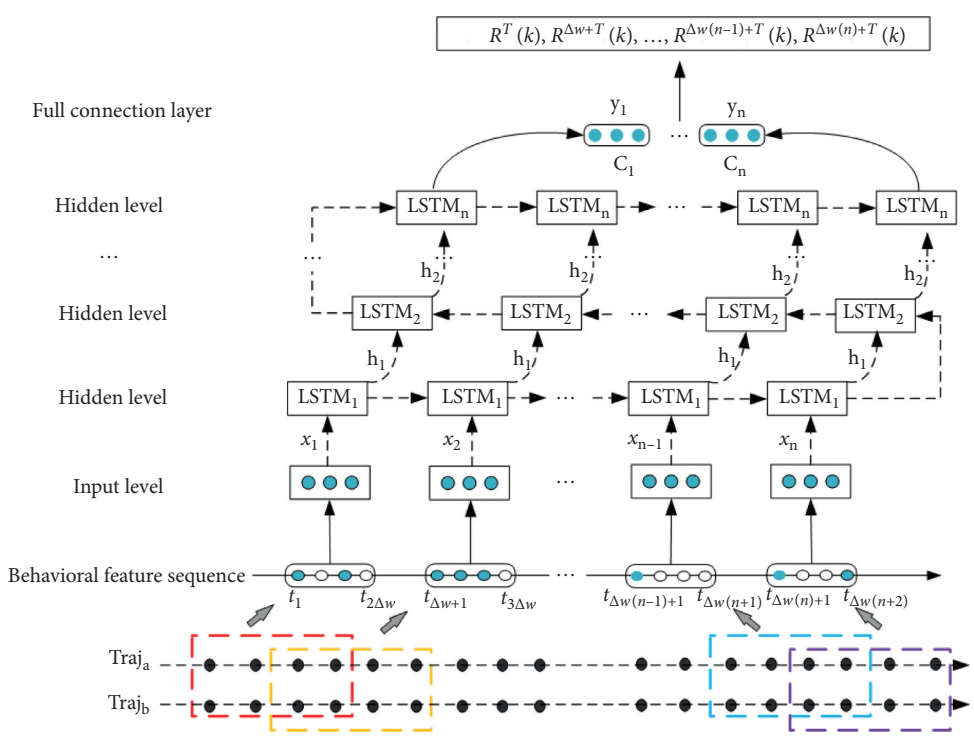


Figure 5: The model framework of ship encounter risk prediction.

constructed by determining the length of time window and sliding step. In our implementation, we set the different window lengths to 20 s and 10 s; thus, there are 185,208 records in the dataset. The records of the individual encounter situations are shown in Table 3. Among them, 159,798 records were employed to train the risk prediction

Figure 6: Electronic chart of the South Channel intersection waterway of the Yangtze Estuary.

Table 3: Records of time windows.

Category of situation	Total	Training set Individual	Validation set Individual	Test set Individual
Crossing situation		45774	5100	2600
Head-on situation	185208	74805	8310	2250
Overtaking situation		39219	4350	2800

model, 17,760 records were employed to determine the optimal parameters of the model, and the remaining records were used for testing. Figure 7 shows the risk level distribution in the test data, thus providing an opportunity to advance our test data knowledge.

5.2. Parameters in the Experiment. This section discusses the various experimental parameters to find an optimal parameter combination to accurately predict the risk of ship encounters. First, we compare different prediction horizons, that is, 30 seconds and 40 seconds. It is reasonable that ship officers have sufficient time to react to emergencies with these prediction horizons. For improving the accuracy of the model, the grid search method is adopted to determine the optimal number of hidden layers and the learning rate of LSTM in the cross-validation set. Figure 8 shows the cross-validation results under the two prediction horizons. In the case of 30 seconds in advance, the peak value of the prediction accuracy is obtained when the hidden layer is 2 and the learning rate is 0.1, and the accuracy is 0.8712. When it is 40 seconds in advance, the optimal number of hidden layers and the learning rate should be 3 and 0.00010, respectively, and the corresponding accuracy is 0.8676. Finally, the number of LSTM units in the individual hidden layer is 18, which is closely related to the six types of parameters in the sequence of behavioral features.

5.3. Experimental Result. In this section, we evaluate the prediction accuracy and robustness in a typical scenario of three encounter situations (crossing, head-on, and

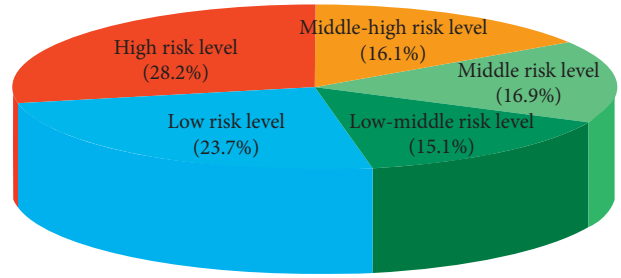


Figure 7: Proportion of risk levels in the test set.

overtaking). In particular, Figures 9–11 display a series of comparisons between the ground truth of the risk level and the predicted results in an individual scenario, and each of them contains eight subgraphs. As in the above process, the ground truth and predicted risk level here both refer to the future risk level corresponding to the current window. Figures 9(a)–11(b) and 9(b)–11(b) show the spatial distribution of the predicted values under various prediction horizons, including 30 seconds and 40 seconds, respectively. Moreover, Figures 9(c)–11(c) present the spatial distribution of the real risk level to evaluate the accuracy of the risk prediction. Furthermore, Figure 9(d)–11(d) and 9(e)–11(e) demonstrate the risk level of each window under the two prediction horizons. It can be observed that the parts above and below the horizontal line are the real risk level and the predicted risk level, respectively. Finally, to study the dynamic change of the risk in the encountering process, we divide the whole encounter process into five stages according to time. Figures 9(f)–11(f) and 9(g)–11(g) are the histograms displaying the predicted risk level of each stage under the three encounter scenarios, and the ratios of individual risk level are intuitively presented in Figure 9(h)–11(h), which are closely related to the prediction accuracy of each stage. The following three typical encounter scenarios are analyzed.

Figure 9 shows the predicted results in the crossing situation and compares them with the actual values. As shown in Figure 9(c), the collision risk is initially at the low risk level, and it is continuing at that level for a while until the ships involved in the encounter sail into the warning zone (area indicated by the red dotted line). Subsequently, the collision risk gradually rises to the high risk level, while one of the ships is in the center of the warning zone, and it commences evasive maneuvers to achieve a safe encounter. Moreover, as shown in Figures 9(a) and 9(b), the risk level predicted by both models is almost inconsistent with the real values at the beginning. With the evolution of the encounter process, there are certain deviations. Later, all the models correctly predict the risk, especially while the collision risk is at a high risk level. It follows that effective predictions can be made by taking advantage of the sequence of behavioral features. Figures 9(d)–9(g) show that the model has the ability to yield more superior predictions, while the prediction horizon is shorter. However, the variation tendencies of true values and predicted values are fairly consistent. As shown in Figure 9(h), the model has achieved high prediction accuracy in general. The proposed approach is

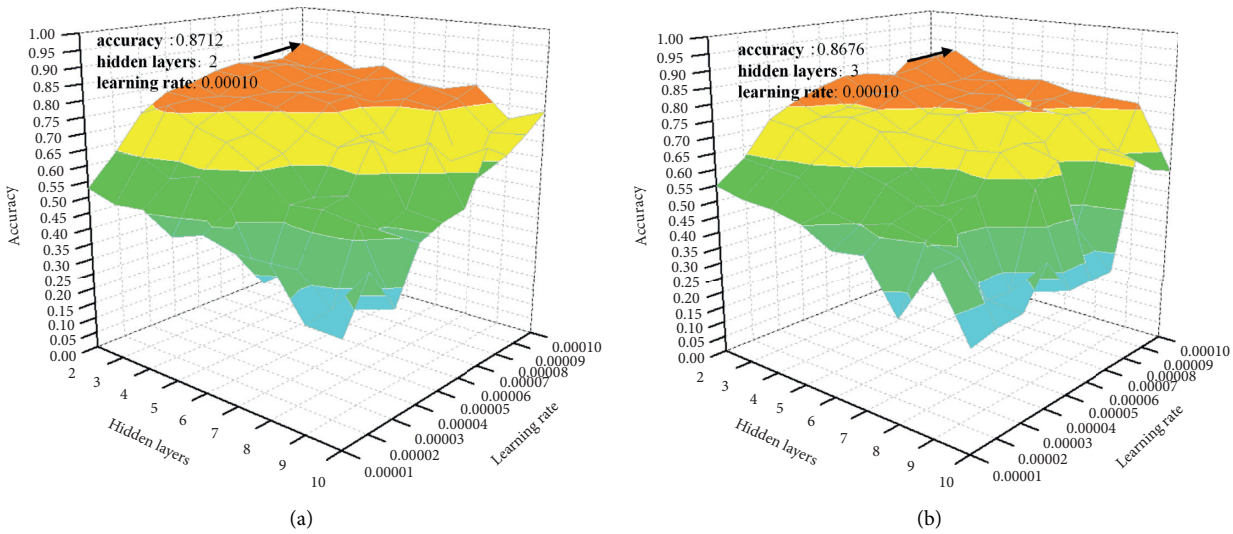


Figure 8: Determining the optimal hidden layers and learning rate of LSTM via a grid search in the validation set.

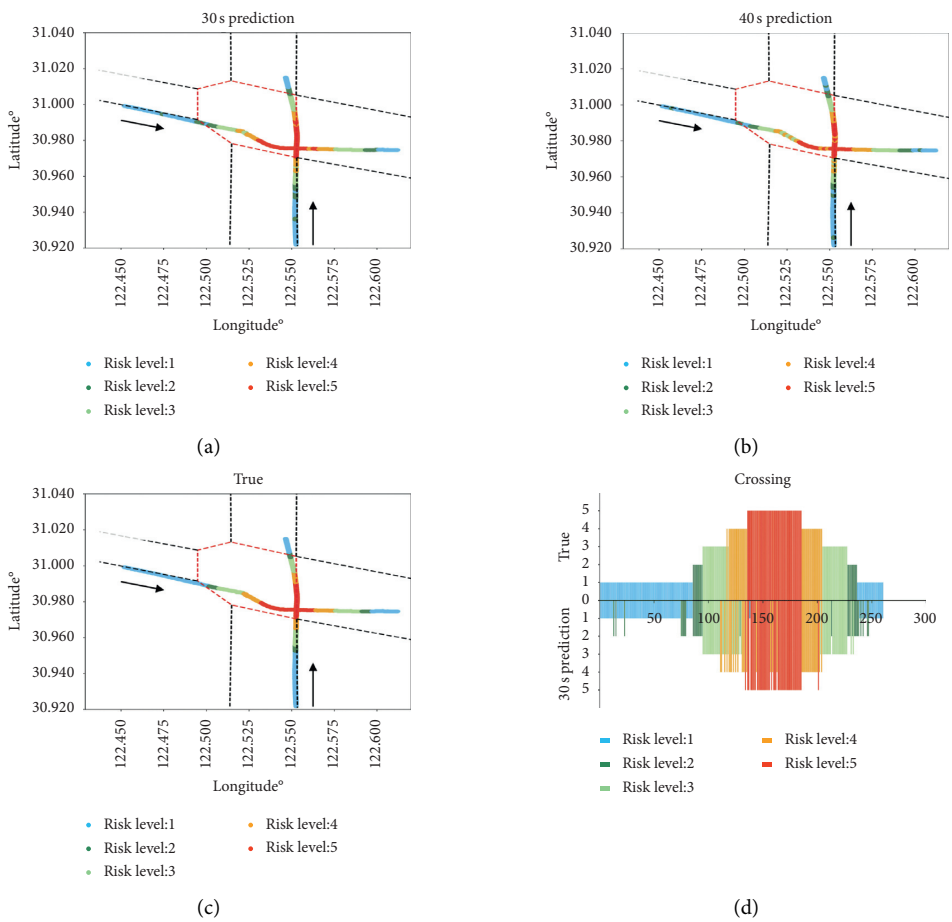


Figure 9: Continued.

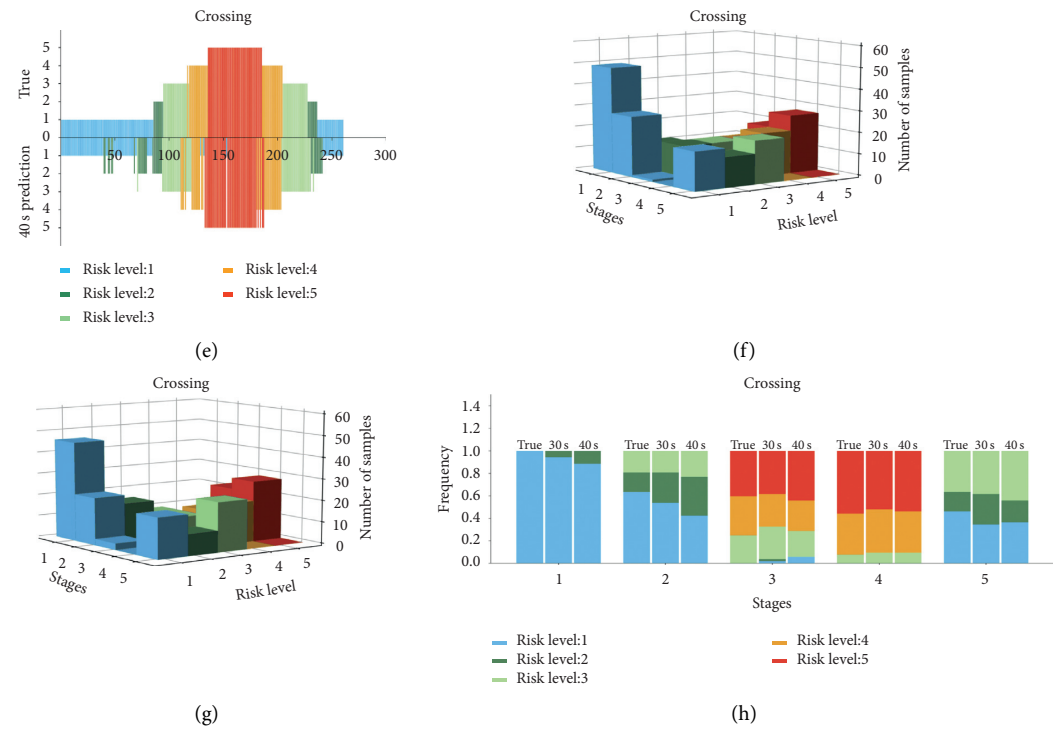


FIGURE 9: Crossing situation: (a) and (b) are the spatial distributions of risk level under various prediction horizons; (c) is the corresponding spatial distribution of realistic risk level. (d) and (e) show the risk level statistics of each window under two prediction horizons. (f) and (g) are the number of samples of each risk level in the five stages. (h) is the frequency of each risk level at various stages.

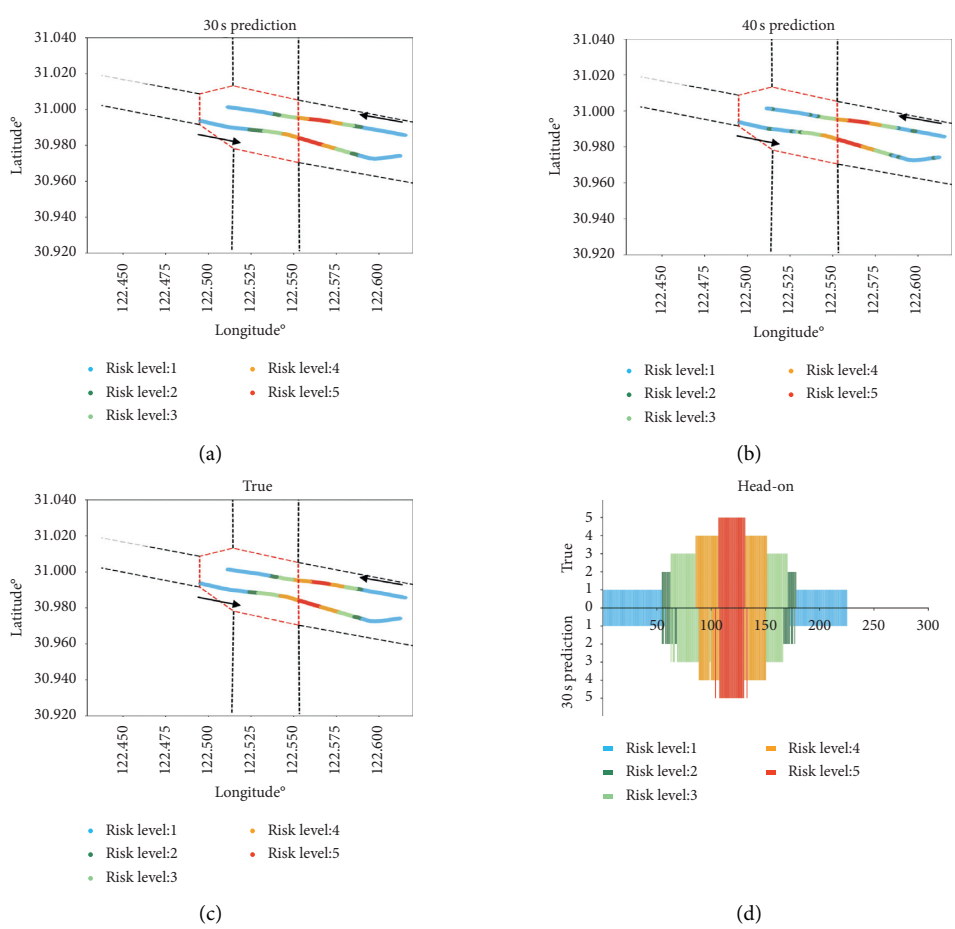


FIGURE 10: Continued.

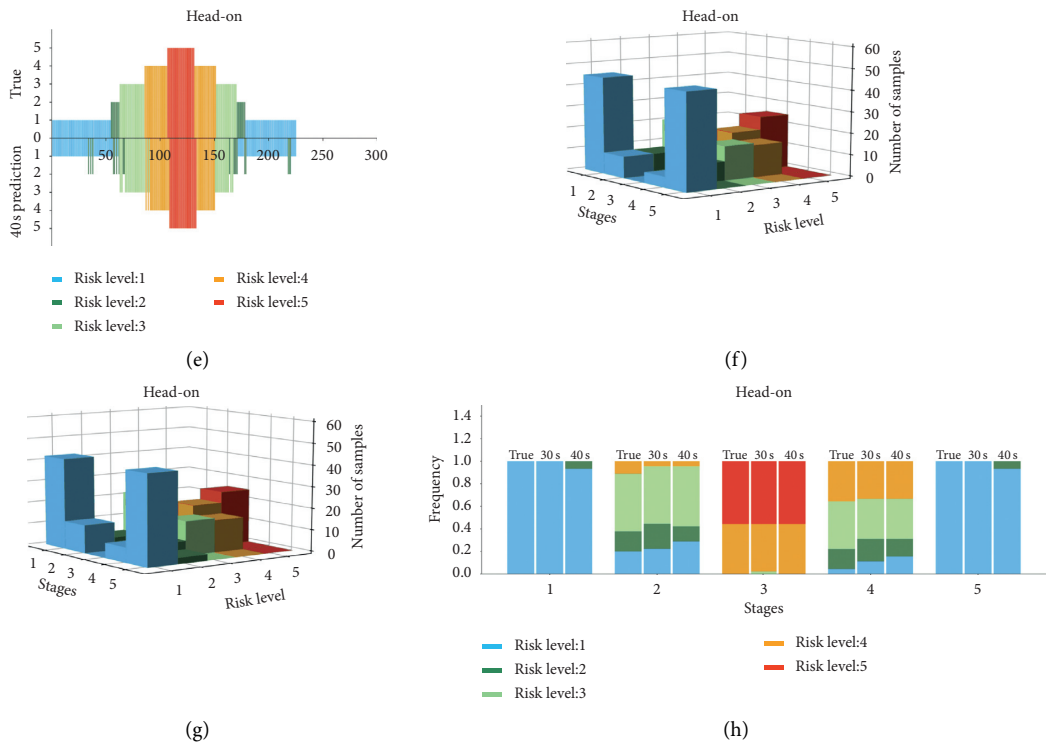


FIGURE 10: Head-on situation: (a) and (b) are the spatial distribution of risk level under various prediction horizons; (c) is the corresponding spatial distribution of realistic risk level. (d) and (e) show the risk level statistics of each window under two prediction horizons. (f) and (g) are the number of samples of each risk level in the five stages. (h) is the frequency of each risk level at various stages.

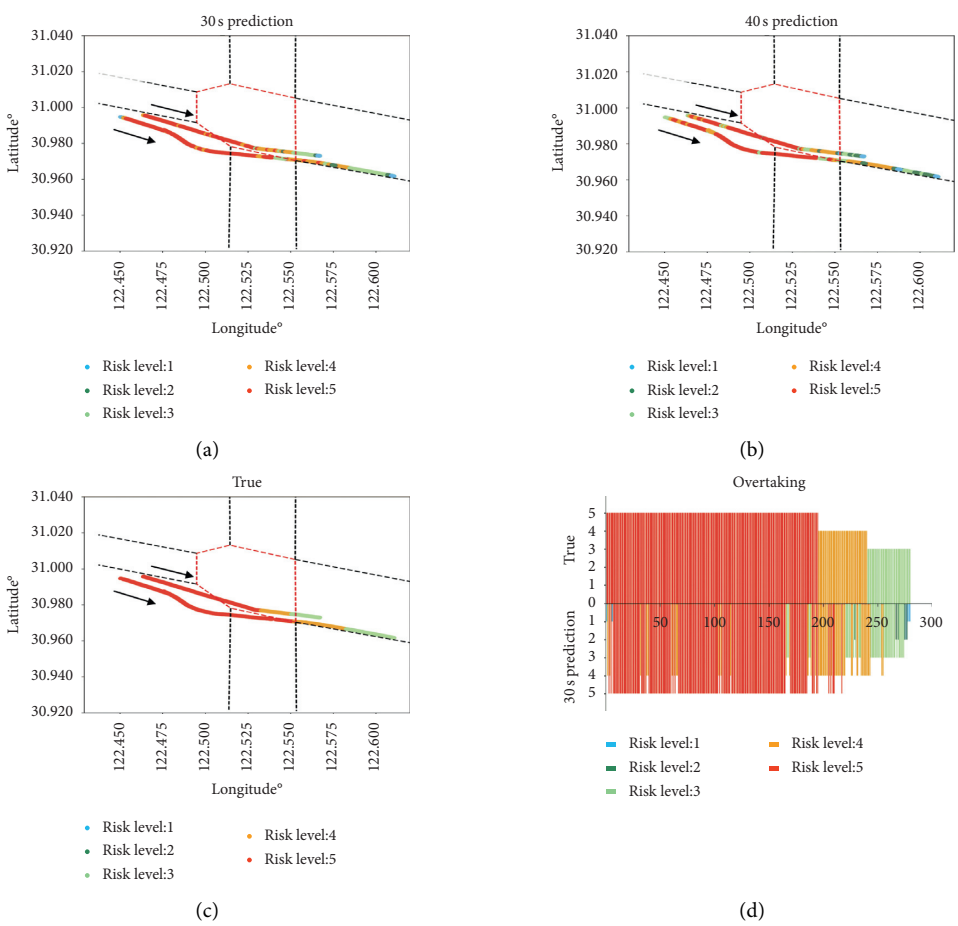


FIGURE 11: Continued.

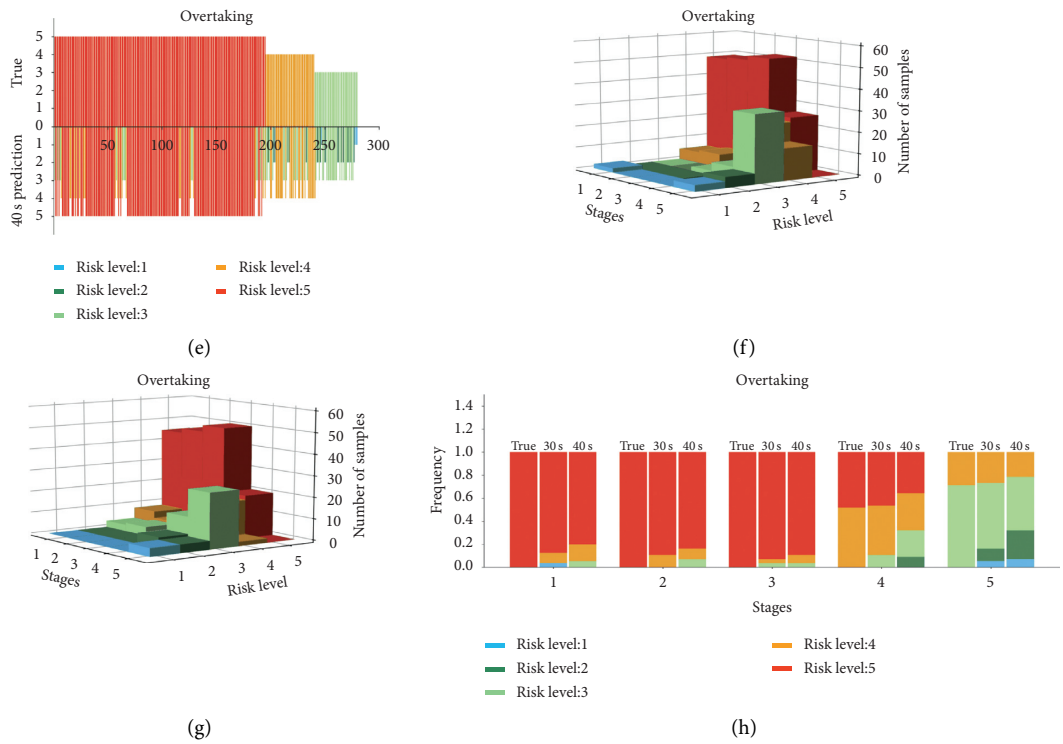


Figure 11: Overtaking situation: (a) and (b) show the spatial distribution of the risk level under various prediction horizons; (c) is the corresponding spatial distribution of the realistic risk level; (d) and (e) show the risk level statistics of each window under the two prediction horizons. (f) and (g) are the number of samples of each risk level in the five stages. (h) is the frequency of each risk level at various stages.

capable of predicting the collision risk under a crossing situation by making full use of the spatiotemporal behaviors of the ships involved in the encounter.

As shown in Figure 10(c), collision risk is initially at a low risk level since the two ships are far apart. As the head-on process evolves, the risk of collision increases gradually. However, since both ships sail in their respective channels, neither of them takes anticollision maneuvers under such a circumstance, although the high risk level has been maintained for a certain period. Eventually, the risk of collision gradually disappears, because the two ships have passed each other (Past and Clear). Moreover, the spatial distribution of risk level in Figures 10(a)and 10(b) is consistent with that in Figure 10(c). In this example, it seems that there is no difference between the predicted results in the 30-second horizon and those in the 40-second horizon. This may be because the motion state of the ship does not change during the course in terms of speed and direction keeping, and the prediction accuracy of the collision risk has only a small relationship with the advance of time. As shown in Figures 10(d)–10(g), it can be observed that the predicted results differ from the actual collision risk, thereby confirming the suitability of our approach in a head-on situation. In particular, accurate prediction of a high risk level is of great significance to avoid potential conflicts in congested waterways because dangerous encounters occur occasionally in these high-risk zones. Therefore, it necessitates additional attention and caution in these areas to ensure safe encounters between ships.

Table 4: Confusion matrix under a prediction horizon of 30 seconds.

Risk level	L	LM	M	MH	H
L	903	7	8	15	18
LM	131	117	63	26	31
M	32	20	468	43	43
MH	18	15	62	449	78
H	16	13	32	82	1135
SUM	1100	172	633	615	1305
ACCURACY (%)	82	68	74	73	87

For the overtaking situation, Figure 11(c) shows that the collision risk has continually been at a high risk level. This is primarily because the relative distance and course between the two ships are small, which makes it easier for a potential collision accident to occur. Soon afterward, the collision risk tends to decline gradually, while one ship leaves the warning zone, marking a safe encounter between two ships as well. The risk distributions with the high risk level in Figure 11(c) are fairly consistent with those in Figures 11(a) and 11(b), which shows that the risk prediction model has a high recall rate for high risk cases. However, for other risk levels, there are certain deviations between the predicted results and the real value. Eventually, we can observe from Figure 11(h) that the model can perceive a high risk from the beginning, which suggests that once the overtaking situation is formed, there is a high risk in the initial stage. In this case, the ship officer can pay additional attention to the possible collision risk according to this early warning model.

Table 5: Confusion matrix under a prediction horizon of 40 seconds.

Risk level	L	LM	M	MH	H
L	893	6	13	21	15
LM	135	118	58	24	27
M	49	19	456	37	59
MH	13	16	72	431	83
H	10	13	34	102	1121
SUM	1100	172	633	615	1305
ACCURACY (%)	81	69	72	70	86

To evaluate the model performance in all the test sets, the comparisons of the overall prediction precision results of the collision risk for the two prediction horizons are shown in Tables 4 and 5. From the overall sample, this model can predict the risk accurately in different horizons. The ability to identify the risk situations can effectively warn the ship officers of potential collisions, which could provide the basis for a navigation decision. Moreover, in terms of the different horizons, the model is more accurate in predicting collision risks that may occur in the near future than in predicting those further away. In practical applications, this model needs to balance the tradeoff between prediction accuracy and horizon length.

6. Conclusions

An AIS data-driven approach has been derived for collision risk prediction in a vessel encounter situation by learning the intership behavior. The approach considers the relationship between intership behavior and future collision risk, which helps to predict the potential collision risk in various encounter situations in advance. To illustrate the approach, the intership behavior is transformed from AIS traces to a sequence of behavioral features by combining a fixed set of parameters. Then, we related the sequence of behavioral features involved in a specified time window to the risk level at a future time; then, the mapping between them was established through an RNN. Furthermore, we tested the approach over encounter cases in the South Channel intersection waterway with various prediction horizons. The prediction results demonstrated that the approach has reasonable and effective ability and that the risk predicted in advance is consistent with the ship encounter situations. In particular, the model has an outstanding ability to identify risk through intership behavior when the potential collision risk is at a high level. This research offers a valuable insight into collision risk prediction by intership behaviors, and the approach is expected to be applied to the implementation of a new collision warning system.

Data Availability

This Data Statement has been confirmed, and there is no further revision by the authors.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant nos. 51679182 and 71874132, the Green Intelligent Inland Ship Innovation Programme, and the Fundamental Research Funds for the Central Universities under Grant 2020-YB-035.

References

[1] Y. He, Y. Jin, L. Huang, Y. Xiong, P. Chen, and J. Mou, "Quantitative analysis of COLREG rules and seamanship for autonomous collision avoidance at open sea," *Ocean Engineering*, vol. 140, pp. 281–291, 2017.

[2] I. Cohen, W.-P. Brinkman, and M. A. Neerincx, "Modelling environmental and cognitive factors to predict performance in a stressful training scenario on a naval ship simulator," *Cognition, Technology & Work*, vol. 17, no. 4, pp. 503–519, 2015.

[3] H. C. Chin and A. K. Debnath, "Modeling perceived collision risk in port water navigation," *Safety Science*, vol. 47, no. 10, pp. 1410–1416, 2009.

[4] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 12, pp. 2218–2245, 2013.

[5] A. Dobrkovic, M. E. Iacob, and J. van Hillegersberg, "Using machine learning for unsupervised maritime waypoint discovery from streaming AIS data," in *Proceedings of the 15th International Conference on Knowledge Technologies and Data-Driven Business*, ACM, Graz, Austria, October 2015.

[6] Z. Xiao, L. Ponnambalam, X. Fu, and W. Zhang, "Maritime traffic probabilistic forecasting based on vessels' waterway patterns and motion behaviors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3122–3134, 2017.

[7] P. Silveira, A. Teixeira, C. Guedes Soares, and T. Santos, *Assessment of Ship Collision Estimation Methods Using AIS Data*, Taylor & Francis Group, London, UK, 2015.

[8] R. Christian and H. G. Kang, "Probabilistic risk assessment on maritime spent nuclear fuel transportation (part II: ship collision probability)," *Reliability Engineering & System Safety*, vol. 164, pp. 136–149, 2017.

[9] J. H. Ahn, K. P. Rhee, and Y. J. A. You, "study on the collision avoidance of a ship using neural networks and fuzzy logic," *Applied Ocean Research*, vol. 37, pp. 162–173, 2012.

[10] C. N. Hwang, J. M. Yang, and C. Y. Chiang, "The design of fuzzy collision-avoidance expert system implemented by H-autopilot," *Journal of Marine Science and Technology*, vol. 9, pp. 25–37, 2001.

[11] J. M. Mou, C. v. d. Tak, and H. Ligteringen, "Study on collision avoidance in busy waterways by using AIS data," *Ocean Engineering*, vol. 37, no. 5-6, pp. 483–490, 2010.

[12] W. Zhang, F. Goerlandt, J. Montewka, and P. Kujala, "A method for detecting possible near miss ship collisions from AIS data," *Ocean Engineering*, vol. 107, pp. 60–69, 2015.

[13] Y. Ren, J. Mou, Q. Yan, and F. Zhang, "Study on assessing dynamic risk of ship collision," in *Proceedings of the First International Conference on Transportation Information and Safety (ICTIS)*, pp. 2751–2757, Wuhan, China, April 2012.

[14] P. A. M. Silveira, A. P. Teixeira, and C. G. Soares, "Use of AIS data to characterise marine traffic patterns and ship collision risk off the coast of Portugal," *Journal of Navigation*, vol. 66, no. 6, pp. 879–898, 2013.

[15] W. Zhang, F. Goerlandt, P. Kujala, and Y. Wang, "An advanced method for detecting possible near miss ship collisions from AIS data," *Ocean Engineering*, vol. 124, pp. 141–156, 2016.

[16] W. Zhang, X. Feng, Y. Qi, F. Shu, Y. Zhang, and Y. Wang, "Towards a model of regional vessel near-miss collision risk assessment for open waters based on AIS data," *Journal of Navigation*, vol. 72, no. 6, pp. 1449–1468, 2019.

[17] R. Szlapczynski, "A unified measure of collision risk derived from the concept of a ship domain," *Journal of Navigation*, vol. 59, no. 3, pp. 477–490, 2006.

[18] R. Szlapczynski and J. Szlapczynska, "An analysis of domain-based ship collision risk parameters," *Ocean Engineering*, vol. 126, pp. 47–56, 2016.

[19] X. Wu, A. L. Mehta, V. A. Zaloom, and B. N. Craig, "Analysis of waterway transportation in Southeast Texas waterway based on AIS data," *Ocean Engineering*, vol. 121, pp. 196–209, 2016.

[20] N. Wang, "An intelligent spatial collision risk based on the quaternion ship domain," *Journal of Navigation*, vol. 63, no. 4, pp. 733–749, 2010.

[21] X. Qu, Q. Meng, and L. Suyi, "Ship collision risk assessment for the Singapore Strait," *Accident Analysis & Prevention*, vol. 43, no. 6, pp. 2030–2036, 2011.

[22] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G. B. Huang, "Exploiting AIS data for intelligent maritime navigation: a comprehensive survey from data to methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 1559–1582, 2017.

[23] G. A. Susto, A. Cenedese, and M. Terzi, "Time-series classification methods: review and applications to power systems data," in *Big Data Application in Power Systems*, pp. 179–220, Elsevier, Amsterdam, Netherlands, 2018.

[24] A. Graves, "Generating sequences with recurrent neural networks," https://arxiv.org/abs/1308.0850.

[25] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1764–1772, Beijing, China, June 2014.

[26] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018.

[27] L. Sun, Z. Yan, S. M. Mellado, M. Hanheide, and T. Duckett, "3DOF pedestrian trajectory prediction learned from long-term autonomous mobile robot deployment data," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[28] L. Gang, Y. Wang, Y. Sun, L. Zhou, and M. Zhang, "Estimation of vessel collision risk index based on support vector machine," *Advances in Mechanical Engineering*, vol. 8, 2016.

[29] J.-B. Yim, D.-S. Kim, and D.-J. Park, "Modeling perceived collision risk in vessel encounter situations," *Ocean Engineering*, vol. 166, pp. 64–75, 2018.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Science & Business Media, Berlin, Germany, 2009.

[31] Z. He, F. Yang, Z. Li, K. Liu, and N. Xiong, "Mining channel water depth information from IoT-based big automated identification system data for safe waterway navigation," *IEEE Access*, vol. 6, pp. 75598–75608, 2018.

[32] T. Macduff, "The probability of vessel collisions," *Ocean Industry*, vol. 9, 1974.

[33] D. Chen, C. Dai, X. Wan, and J. Mou, "A research on AIS-based embedded system for ship collision avoidance," in *Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS)*, pp. 512–517, IEEE, Wuhan, China, June 2015.

[34] F. Kaneko, "Methods for probabilistic safety assessments of ships," *Journal of Marine Science and Technology*, vol. 7, no. 1, pp. 1–16, 2002.

[35] Y. Koldenhof, C. Van der Tak, and C. Glansdorp, "Risk Awareness; a model to calculate the risk of a ship dynamically," in *Proceedings of the XIII International Scientific and Technical Conference on Marine Traffic Engineering*, pp. 112–119, Malmö, Sweden, October 2009.

[36] H. Yixiong, Y. Xiong, H. Liwen et al., "Studies of last steering point/CRI basis on MMG and ship domain," *Journal of Wuhan University of Technology: Transportation Science and Engineering*, vol. 38, pp. 1088–1091, 2014.

[37] A. Zyner, S. Worrall, and E. Nebot, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1759–1764, 2018.

[38] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, "Traffic speed prediction under non-recurrent congestion: based on LSTM method and BeiDou navigation satellite system data," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 2, pp. 70–81, 2019.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8847752, 13 pages
https://doi.org/10.1155/2020/8847752

Research Article

Machine Learning Approach to Quantity Management for Long-Term Sustainable Development of Dockless Public Bike: Case of Shenzhen in China

Qingfeng Zhou,[1,2] **Chun Janice Wong**,[1] **and Xian Su**[3]

[1]*Harbin Institute of Technology, Shenzhen, Guangdong 518055, China*
[2]*Shenzhen Key Laboratory of Urban Planning and Decision Making, Shenzhen, Guangdong 518055, China*
[3]*China Resources Land Guangxi Co, Nanning, Guangxi 530000, China*

Correspondence should be addressed to Chun Janice Wong; janicewong@hit.edu.cn

Received 21 July 2020; Revised 25 October 2020; Accepted 12 November 2020; Published 28 November 2020

Academic Editor: Petr Dolezel

Since the number of bicycles is critical to the sustainable development of dockless PBS, this research practiced the introduction of a machine learning approach to quantity management using OFO bike operation data in Shenzhen. First, two clustering algorithms were used to identify the bicycle gathering area, and the available bike number and coefficient of available bike number variation were analyzed in each bicycle gathering area's type. Second, five classification algorithms were compared in the accuracy of distinguishing the type of bicycle gathering areas using 25 impact factors. Finally, the application of the knowledge gained from the existing dockless bicycle operation data to guide the number planning and management of public bicycles was explored. We found the following. (1) There were 492 OFO bicycle gathering areas that can be divided into four types: high inefficient, normal inefficient, high efficient, and normal efficient. The high inefficient and normal inefficient areas gathered about 110,000 bicycles with low usage. (2) More types of bicycle gathering area will affect the accuracy of the classification algorithm. The random forest classification had the best performance in identifying bicycle gathering area types in five classification algorithms with an accuracy of more than 75%. (3) There were obvious differences in the characteristics of 25 impact factors in four types of bicycle gathering areas. It is feasible to use these factors to predict area type to optimize the number of available bicycles, reduce operating costs, and improve utilization efficiency. This work helps operators and government understand the characteristics of dockless PBS and contributes to promoting long-term sustainable development of the system through a machine learning approach.

1. Introduction

The public bike system (PBS) also called a bicycle sharing system (BSS), which was born in 1965 in Europe, has been developed for three generations [1]. PBS is economical, eco-friendly, healthy, more equitable, produces ultralow carbon emissions, and has rapidly emerged in many cities all over the world [2]. Since 2016, a relatively new model of PBS, known as the free-float bike sharing system, has increasingly gained its popularity. The FFBS is based on the mobile app and GPS which eliminates stations and docks (also called dockless bike). Passengers can easily pick up and drop off a bike anywhere using their cell phones. This system is quite spread nowadays through enterprises as OFO and Mobike since early 2016 in China. Dockless PBS brings new experiences and conveniences as well as some problems, and an important issue is to consider the number of bicycles available. It has two sides:(1) assuming that the surrounding roads are suitable for cycling when a large number of dockless bicycles are concentrated in an area with a low cost to use, we can think it is providing "adequate" bike supply, which can help us fully understand the bike demand in this area; (2) in fact, if the number of available bicycles is too large, it will cause a series of waste. Many problems are related to the number of shared bicycles especially for the dockless PBS which is an important issue to be considered. But it is seldom involved in the existing research. The number of available bicycles is the core indicator. Excessive

bicycles can affect the cost and efficiency of operation, which is not conducive to the long-term sustainability of the system. The government and scholars have paid more and more attention to the question of how to rationally develop dockless PBS in the city.

Computational intelligence, such as artificial neural networks, fuzzy systems, and evolutionary computing, has achieved significant results in modeling, learning, and search and optimization problems for smart city applications [3, 4]. The characteristics of machine learning make it attractive for analyzing smart city data with complex nature [5], such as modes (streams, time series, images, videos, and texts), large amounts (continuous data generated by millions of sensing devices), space-time dependence, etc. Researchers in smart cities have applied machine learning in many areas, such as urban human mobility [6], public space utilization [7], and public bus charging station placement [8]. Since the number of bicycles is critical to the sustainable development of dockless PBS, this research practiced the introduction of a machine learning approach to quantity management. Four issues are discussed from the existing shared bicycle operation data. (1) How to identify the gathering area of dockless shared bicycles? (2) How to measure the number of bicycles and activity characteristics in the bicycle gathering area? (3) What are the differences between classification algorithms in predicting the types of bicycle gathering areas? (4) How to use activity pattern to guide dockless PBS rationally develop in the city? In this study, first, two clustering algorithms were used to identify the bicycle gathering area, and the available bike number and coefficient of available bike number variation were analyzed in each bicycle gathering area's type. Second, five classification algorithms were compared in the accuracy of distinguishing the type of bicycle gathering areas using impact factors. Finally, the application of the knowledge gained from the existing dockless bicycle operation data to guide the number planning and management of public bicycles was explored.

The rest of this paper is organized as follows. Section 2 presents a literature review on the systems perspective of public bike research. Section 3 introduces the indicators and methods used. Section 4 briefly describes bicycle operation data and influencing variables. In Section 5, we discuss the bicycle gathering area type's recognition, prediction, and application. Finally, Section 6 summarizes the results of this study and provides direction in future studies.

2. Previous Work

PBS is involved in many areas of research, and it is broadly based on two perspectives: user perspective and systems perspective [9]. In this study, we only focus on a systems perspective according to the goals.

2.1. Bike Sharing Rebalance. For PBS, the lack of resources is the major issue: a user can arrive at a station that has no bike available or wants to return her bike at a station with no empty spot. Based on the practical usage, several studies focused to deal with public bike rebalancing problem using intelligent algorithms. Fricker and Gast [10] proposed a stochastic model of homogeneous PBSs to study the effect of users' random choices on the number of problematic stations. They also computed the rate of which bikes must be redistributed by trucks to ensure a given quality of service. You et al. [11] provided an integrated model to resolve the problems of fleet sizing, empty-resource repositioning, and vehicle routing for bike transfer in multiple station systems. O'Mahony and Shmoys [12] tackled the problem of rebalancing PBS during rush hour. An optimization problem whose goal is to plan truck routes to make PBS as balanced as possible in night shift was studied, and novel methods were developed for optimizing rebalancing resources. Chen et al. [13] addressed the layout planning of public bicycle system within the attracted scope of a metro station. Locations of different PBS service stations and the optimal route options for the implementation of the redistributing strategy were considered. Lozano et al. [14] proposed a multiagent model that provides visualization and prediction tools for PBS.

2.2. Bike Demand Estimation. These studies examine the influence of PBS infrastructure, transportation network infrastructure, land use and urban form, meteorological data, and temporal characteristics on PBS usage. Faghih-Imani et al. [15] collected station-level occupancy data and then transformed station occupancy snapshot data into station-level customer arrivals and departures. They developed a mixed linear model to estimate the influence of bicycle infrastructure, sociodemographic characteristics, and land-use characteristics on customer arrivals and departures. In the work of Krykewycz et al. [16], various demographic, land use, and infrastructure factors understood to be favorable for bike share usage were spatially analyzed to define a primary market area. El-Assi et al. [17] investigated the effects of weather, socioeconomic and demographic factors, and land use and the built environment on bicycle share ridership. A regression analysis was performed on three different levels. Hampshire and Marla [18] employed a panel regression model to explain the factors affecting the bike sharing trip generation and attraction in the presence of unobserved spatial and temporal variables. The data used included PBS's usage data in Barcelona and Seville, nine census demographic data, and the location of points of interest (POIs). Zhang et al. [19] employed a multiple linear regression model to examine the influence of built environment variables on trip demand as well as on the ratio of demand to supply at bike stations in China. Faghih-Imani et al. [15] investigated factors affecting bicycle share demand at the station level using real-time ridership data. The results showed that stations close to major roads had lower trip activities compared to stations that were situated around minor roads and bicycle lanes. A number of land use and built environment variables, temporal characteristics, and weather variables such as temperature were investigated. Maurer [20] used a pairwise suitability analysis to understand the effects of variables such as job density, household income, and alternative commuters on public bicycle share ridership to propose the locations of bicycle stations in

Sacramento, California. Gebhart and Noland [21] used real-time ridership data for Capital Bikeshare in Washington D.C. to investigate the impact of weather variables and proximity of bike share stations to metro stations on ridership levels. Buck and Buehler [22] investigated the influence of bicycle infrastructure, population density, land use mix around stations, and the number of households without a car using bicycle share systems using ridership data from Capital Bikeshare. Wang et al. [23] evaluated the effect of sociodemographic, land use, built environment, and transportation infrastructure variables on bicycle share ridership. Rixey [24] explored the influence of sociodemographic characteristics such as education, income, and employment and population density on monthly ridership data from three states of the USA.

2.3. Spatial and Temporal Patterns of Bike Use. These studies explore the spatial and temporal patterns of bike use over the time of day, using data mining and visualization techniques. Clustering is frequently used to identify mobility patterns in BSS usage by partitioning the stations into different clusters having a similar usage. Wong and Cheng [25] presented the insights of imbalanced public bicycle distributions through the analysis of spatiotemporal activity patterns of bike stations. The clustering algorithm was used to analyze how station activity patterns were geographically distributed based on their usage patterns. They also explored how these activity patterns relate to underlying cultural and spatial characteristics of Taipei City in China. Temporal and spatiotemporal patterns among bike stations of Barcelona bike sharing system were explored by Froehlich et al. [26]. Numerous research studies also used a hierarchical clustering method to generate clusters and investigate usage patterns geographically distributed in the city to understand the impact of the inhomogeneity of the city on the long-run activity of stations [27–29]. Brien et al. [30] proposed a classification of bike shares based on the geographical footprint and diurnal, day-of-week, and spatial variations in occupancy rates. Etienne and Latifa [31] presented an automatic algorithm based on a new statistical model to automatically cluster PBS stations according to their usage profile. Zhou [32] investigated the spatiotemporal biking pattern in Chicago by analyzing massive BSS data from July to December in 2013 and 2014. Bike flow similarity graph was constructed with a fast greedy algorithm to detect spatial communities of biking flows.

Scholars have achieved rich results in measuring the indicators of the bicycle system and the factors affecting cycling. The methods of research mainly involve regression models. The knowledge gained most comes from the dock PBS except a few studies [33–35].

3. Methodology

3.1. Indicators of Dockless PBS. There are many indicators to measure the PBS, including the number of bicycle use, arrival rate, and departure rate. This study focused on the number of available bicycles and their changes, so two indicators were used.

3.1.1. Average Available Bike Number. Unlike the dock PBS, the maximum available bike number is fixed and determined by the number of docks of station. For the dockless PBS, the maximum available bike is not subject to parking restrictions. It is related to the initial bike quantity status of deployment by the system and varies as the bike flows. We proposed the average available bike number to explore the dockless PBS. It represents the number of bicycles available in a bike service area. This metric is used to measure the bicycle resource. The number of bikes available per hour (Abn_i) can be calculated by equation (1).

$$abn_i = \frac{\sum_{d=1}^{5} abn_d^i}{5}, \tag{1}$$

$$abn\_\mathrm{DAY} = \frac{\sum_{i=1}^{24} Abn_i}{24}, \tag{2}$$

where i represents the ith hour in a day; d represents the dth workday of a week; Abn_d^i is the number of available bicycles in the ith hour of the dth day; Abn_i is the average number of available bicycles in hour i in work day; and abn_DAY in equation (2) is the average available vehicle for bike service area throughout the day.

3.1.2. Coefficient of Available Bike Number Variation. The coefficient of variation is used to compare the degree of dispersion of the two sets of data, which can eliminate the influence of measurement scale and dimension. In this study, the coefficient of variation was used to compare the changes in available bicycles in 24 hours of a day among bicycle service areas. The calculation formula is shown in the following equation:

$$cv = \frac{\mathrm{s.d}\left(abn_i\right)}{abn\_\mathrm{DAY}} \times 100, \tag{3}$$

where $\mathrm{s.d}\left(abn_i\right)$ is the standard deviation of available bicycles number in 24 hours in a service area. Obviously, cv is affected by the two statistics of mean and standard deviation of available bike number. This metric is used to measure the variation in bicycle usage with average available bike number.

3.2. Clustering and Classification in Machine Learning Approach

3.2.1. Clustering Algorithm. Clustering is an unsupervised learning algorithm of classifying and organizing members in datasets which are similar [36].

(1) k-Means Clustering Algorithm. Given a set of data, the k-means algorithm divides the data into k clusters repeatedly according to a distance function. The algorithm operates on a set of d dimensional vectors, $D = \{x_i | i = 1, \ldots, N\}$, where $x_i \in d$ denotes ith data point. The algorithm is initialized by picking k points in d as the initial k cluster representatives or "centroids." Techniques for selecting these initial seeds include sampling at random from the dataset, setting them as

the solution of clustering a small subset of the data, or perturbing the global mean of the data k times. Then, the algorithm iterates between two steps till convergence. About the value of k, we can choose from reasonable guessed or predefined number, but it is better to know whether k clusters is better or worse than $k-1$ or $k+1$ clusters. The method of With he Sum of Square (WSS) is often used to get reasonable K value. WSS is the sum of the square of the distance between all points and their nearest centroid point. The calculation is shown in equation (4). p_i represents the point i, and q^i represents the nearest centroid point to i; if all data points are relatively close to their respective centers, then the WSS is relatively small. If $K+1$ clusters do not significantly reduce the WSS value of K clusters, then the classification is of little significance.

$$\mathrm{WSS} = \sum_{i=1}^{N} d\left(p_i, q^i\right)^2. \tag{4}$$

(2) Mean Shift Clustering Algorithm. Mean shift clustering is a general nonparametric cluster finding procedure introduced by Fukunaga and Hostetler [37], and it does not depend on any explicit assumptions on the shape of the point distribution, the number of clusters, or any form of random initialization. Mean shift treats the clustering problem by supposing that all points given represent samples from some underlying probability density function, with regions of high sample density corresponding to the local maxima of this distribution. To find these local maxima, the algorithm works by allowing the points to attract each other, via what might be considered a short-ranged "gravitational" force. Allowing the points to gravitate towards areas of higher density, one can show that they will eventually coalesce at a series of points, close to the local maxima of the distribution. Those data points that converge to the same local maxima are considered to be members of the same cluster. For a mathematical details, see Comaniciu and Meer [38]. In the next sections, we illustrate application of the algorithm to a couple of problems using the python package SkLearn which contains a mean shift implementation.

3.2.2. Classification Algorithm. Classification is a kind of supervised learning algorithm of training a classifier in a group of samples that already know the class label so that it can classify an unknown sample. In the field of machine learning, there are hundreds of classifiers to solve real-world classification problems [39], and in this research, five commonly used classification algorithms are selected: random forest classifier, K-nearest neighbor classifier, logistic regression, support vector machine, and artificial neural network. The five algorithms used in this study are based on the Python platform Scikit-learn package free from https://scikit-learn.org. The parameters of each algorithm have been adjusted to ensure the optimal performance of the algorithm. In the analysis of shared bicycles, the accuracy and robustness of these five classification algorithms will be compared.

(1) Random Forest Classifier. Random forest classifier (RFC) is the most widely used supervised machine learning algorithm. It is very powerful and usually gives good results without the need to repeatedly adjust the parameters. The basic unit of random forests is the decision tree. A random forest is a classifier that contains multiple decision trees, and the category of its output is determined by the mode of the category of the individual tree output [40]. For an input sample, N trees will have N classification results. The random forest integrates all the classification voting results and specifies the category with the highest number of votes as output. It has several advantages: it enables to handle thousands of input variables without variable deletion and gives estimates of what variables are important in the classification.

(2) K-Nearest Neighbor Classifier. K-nearest neighbor (KNN) is a method of measuring the distance between different feature values for classification. Given a training set D and a test object z, the test object is a vector composed of attribute values and an unknown category label. The algorithm needs to calculate the distance (or similarity) between z and each training object. In this way, the list of nearest neighbors can be determined. Then, assign the category with the dominant number of instances in the nearest neighbor to z. The advantage is that it is easy to understand, and good performance can be obtained without excessive adjustment. The disadvantage is that the prediction speed is slow and the dataset with many characteristics cannot be processed. It is vulnerable to data imbalance. And the interpretability of the output is not strong.

(3) Logistic Regression. Logistic regression (LR) is essentially a linear classifier, which refers to the establishment of a regression formula on the classification boundary line based on the existing data to classify. The calculation cost of this method is not high, and it is easy to understand and implement. The fitted parameters can clearly see the impact of each feature on the result. And most of the time is used for training, and classification is fast after training is completed, but it is easy to underfit and the classification accuracy is not high. The main reason is that LR is linear fitting, but in reality, many things do not satisfy linearity.

(4) Support Vector Machine. Support vector machine (SVM) maps the data to a multidimensional space in the form of points, thereby converting the nonlinear separable problem in the original sample space into a linear separable problem in the feature space so that the optimal hyperplane for classification can be found. Then, classify the set according to the hyperplane. SVM can make good predictions on data outside the training set and has a low generalization error rate, low computational overhead, and easy-to-interpret results, but it is too sensitive to parameter adjustments and kernel function parameters.

(5) Artificial Neural Network. Artificial neural network (ANN) is an information processing system based on imitating the structure and function of the brain's neural

network. The ANN algorithm is a set of continuous input/output units, where each connection is associated with a weight. In the learning stage, by adjusting the weights of the neural network, the correct class label of the sample to be learned can be predicted. The advantages of the ANN algorithm are high classification accuracy and strong distributed parallel processing capabilities. Artificial neural networks have strong robustness and fault tolerance for datasets containing a large amount of noisy data, but the learning process cannot be observed, and the output results are difficult to interpret, which will affect the reliability and acceptability of the results. It also requires a large number of parameters, such as network topology, initial values of weights, and thresholds.

4. Study Area

4.1. OFO Dockless PBS in Shenzhen. This paper focuses on China's fastest urbanizing city, Shenzhen, to lay a foundation for empirical analysis of the intensity of usage of the OFO dockless bike sharing system. It provides a unique case study as it is one of the largest bike share programs located in a metropolis. OFO bicycle sharing system was launched in Shenzhen in December 2016 with more than 2200,00 bicycles. We scanned the working status of these bicycles every 15 minutes in one week of September 2017. There are about 57.6 million bicycle status records in a day. For a bicycle ID, we first judge whether the bicycle is used by comparing whether its position has changed. If changed, we saved the time and position of the bicycle. Then, according to the average travel speed and travel distance of the bicycle, the abnormal bicycle use record is rejected. Figure 1 demonstrates the bike service area in Shenzhen.

Figure 2 shows the trip summary of shared bicycles in 24 hours in a workday. There are two distinct peaks in shared bicycle use in a workday. The morning peak is between 08:00-09:00, and the evening peak is between 18:00–21:00. It is reasonable to assume that bikes are used for commuting. In the morning peak, the trip number of bicycles exceeded 50,000. The trips in evening peak were slightly lower than the early peak, but still more than 40,000. During the period of 01:00–06:00, bicycle usage is stable and the lowest with about 5,000 trips per hour. At noon period from 12:00 to 15:00, the bicycle use is about 20,000 per hour. The amount of bicycle use dropped from 40,000 to 10,000 per hour at the night period which is from 22:00 to 24:00.

4.2. Influencing Factor of Bike Use. In the previous studies, factors influencing public bike usage are grouped into four categories: transportation, land-use/build environment, population, and meteorological data. The weather variables are not considered in our study. A total 25 factors were selected including 6 categories of variables: population, point of interest (POI), road network, public transportation, distance, and building function. The detailed factors are listed in Table 1.

5. Results and Discussion

5.1. The Identification of Bicycle Gathering Area. We used the mean shift clustering method to identify the clustering area of bicycles based on the position of the bicycle at 09:00. In the choice of bandwidth, we considered two bandwidths: 300 meters and 500 meters, because the area identified by these two bandwidths is approximately equal to the grid area size of $500\,\mathrm{m} * 500\,\mathrm{m}$ and $1000\mathrm{m} * 1000\,\mathrm{m}$. The minimum number of bicycles included in each category is set to 100. When the bandwidth is 300 meters, a total of 492 bicycle gathering areas are obtained. The 492 bicycle gathering areas contain a total of 140,000 bicycles, accounting for 63.6% of all bicycles. When the bandwidth is 500 m, a total of 270 gathering areas are obtained, including 140,000 bicycles. Considering that the bicycles contained in the 492 clusters are more compact, we finally selected 492 clusters as the analysis objects.

Figure 3 shows the OFO bicycle gathering area identified by mean shift clustering. In Figure 3, each cluster has a center point and the buffer analysis was proposed to obtain the range of the bicycle gathering area. The buffer is a kind of influence range or service scope of the geospatial target, which refers to the polygons of a certain width which are automatically established around the point, the line, and the surface entity. 300 meters of buffers were established by ArcGIS based on all cluster center points, thus to calculate the indicators of dockless PBS and influencing factor of bicycle gathering area.

5.2. Performance of Five Classification Algorithms. After calculating the available bike number and coefficient of available bike number variation of bicycle cluster area, the k-means algorithm was executed to group these areas. Figure 4 shows the WSS curve, and we made WSS values from 2 clusters to 19. When k is increased from 2 to 8, WSS decreases significantly. When $k > 8$, the improvement of WSS is very linear so the cluster centers have similar characteristics. The larger the k means the more the classifications of bicycle cluster area which is likely to impact on the accuracy of the classification algorithm. It is necessary to find an optimal k value to balance between the accurate cluster and accurate classification prediction. This research adopts an experimental strategy to select k from 3 to 8 and then uses five classification algorithms to compare the prediction accuracy. The bicycle gathering areas in the same cluster are marked with the same label using k-means clustering. Five classification algorithms were compared in the accuracy of distinguishing the type of bicycle gathering areas using 25 impact factors. The experimental process is divided into two stages including training and application. At the stage of training, the 492 gathering areas are randomly divided into two parts. The first part containing 75% of areas is used for training data, and the second part as the test data is used to verify the accuracy. Figure 5 shows the accuracy of the five classification algorithms in the training set and the test set when k takes different values.

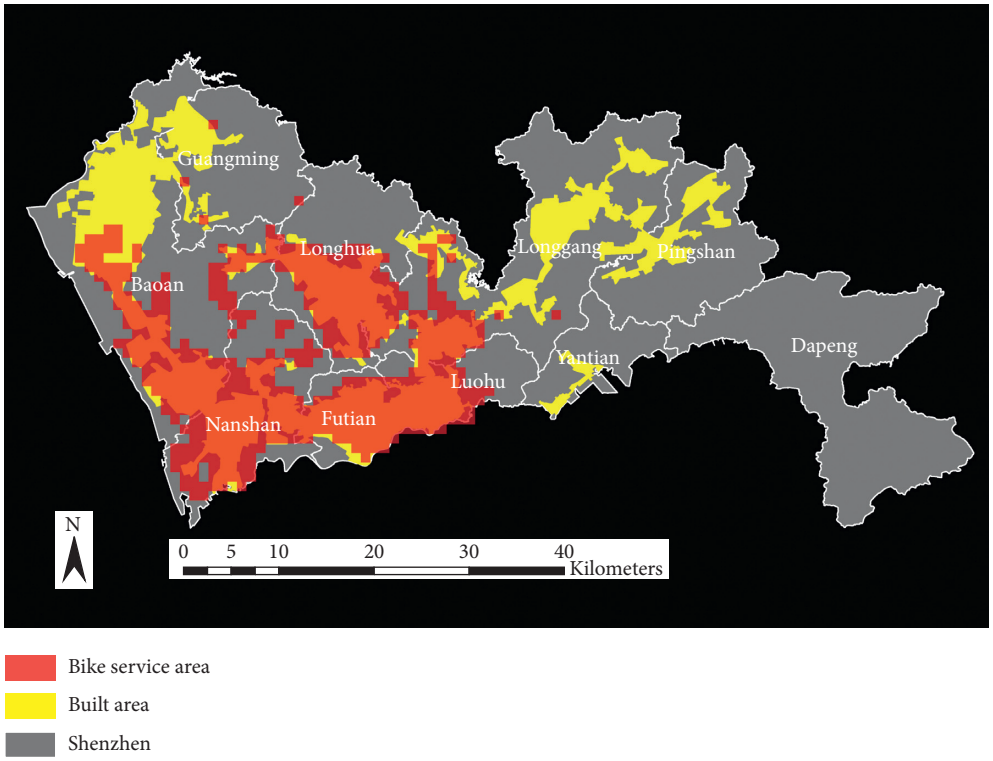


Figure 1: OFO bike service area in Shenzhen.

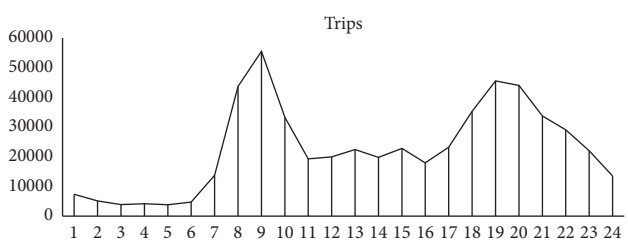


Figure 2: The number of sharing bicycle trips in 24 hours in a workday.

In the training set, the performance differences of the five algorithms are obvious. For different K values, the RFC always maintains the highest accuracy rate, which is higher than 90%. The ANN also has a high accuracy rate. When K is 3-4, the accuracy rate is above 90%, and when the k value is 5–8, the accuracy rate drops to above 80%. KNN algorithm performance is in the middle of the five algorithms. When the K value is 3-4, the accuracy rate is above 70%, and when the K value is 5–8, the accuracy rate drops to above 60%. As the k value increases, the accuracy of SVM drops from 63% to 48%. The worst-performing algorithm is the LR algorithm. As the k value increases, the accuracy rate drops from 58% to 37%. In addition, in the trend that the accuracy rate changes with the value of k, the accuracy rate of the RFC algorithm fluctuates little, and the other algorithms have the highest accuracy rate when the value of K is small. As the value of k increases, the accuracy rate decreases, and When $k = 8$ with ANN, the accuracy rate has increased. In the test set, the accuracy of the five algorithms is lower than that of the training set. The most accurate performance is still the RFC algorithm. When $k = 4$, RFC has the highest accuracy rate of 76.97%, which is the only case in the test set where the accuracy rate exceeds 75%. When $k = 3$, its accuracy rate is 73%. For ANN, the highest accuracy rate is 71% when $k = 4$. The accuracy of KNN in the training set is better than that of SVM and LR, but its performance in the test set is not much different from SVM and LR. The accuracy of these three algorithms is low. In addition, RFC, ANN, and LR all have the highest accuracy when $k = 4$.

Comprehensive comparison of the performance of the five algorithms in the training set and the test set shows that RFC and ANN have better performance in the prediction of the type of bicycle clusters. The accuracy of ANN, SVM, and LR in the training set is quite different, but the difference in the test set is not obvious. Basically, when the k value is greater than 4, as the K value increases, the accuracy of the five classification models has a downward trend. When $K = 4$, RFC and ANN have the highest test accuracy. We choose $k = 4$, which means that the bicycle aggregation is divided into 4 types for further analysis.

5.3. The Analysis of Bicycle Gathering Area

5.3.1. The Clustering of Bicycle Gathering Area. Table 2 shows the description of the cluster center when $k = 4$. The table mainly lists four indicators, which are the standard and original values of Abn_DAY, and the standard and original values of the coefficient of variation (in k-means clustering, standard values were used).The four cluster centers have obvious characteristics. We first divide

Table 1: The influencing factors of bike use.

Factor type	Factor	Unit	Calculation
Population	Population	Number	Count the number of residents in the service area
POI	Restaurant	Number	Count the number of POIs of the corresponding category in the service area
	Company	Number	
	Small store	Number	
	Car park	Number	
Road network	Length of main road	m	Calculate the total length of the corresponding road level in the service area
	Length of secondary road	m	
	Length of branch road	m	
Public transportation	Bus stop	Number	Calculate the number of bus stops in the service area
	Distance to subway	m	Calculate the distance from the center of the service area to the nearest subway station
Distance	Distance to university	m	Calculate the distance from the center of the service area to the closest corresponding place
	Distance to government	m	
	Distance to supermarket	m	
	Distance to hub	m	
	Distance to square	m	
	Distance to park	m	
	Distance to school	m	
	Distance to hospital	m	
Building function	Office building	m^2	Calculate the total floor area of the corresponding building in the service area
	Industrial building	m^2	
	Public building	m^2	
	Commercial building	m^2	
	Residential building	m^2	
	Urban village building	m^2	
	Warehouse	m^2	
	Building number	Number	
	Cover ratio	%	The ratio of the projected area of all buildings to the area of the service area

the four clusters into inefficient and efficient groups according to the value of *cv*. A high *cv* indicates that the number of available bicycles in the gathering area is more appropriate, and the use of regional bicycles is efficient. A low cv indicates that the number of available bicycles in the gathering area is large, which does not match the number of active bicycles, and the use of bicycles is inefficient. We call clusters with z_*cv* lower than 0 as an inefficient group and clusters with z_*cv* greater than 0 as an efficient group. Then, each group is divided into two subtypes according to *Abn*_DAY and *cv*.

(i) Cluster A: $z\_Abn\_\mathrm{DAY} > 1$, $z\_cv < 0$: this cluster can be called high efficiency mode. *Abn*_DAY in this group is reaching 416, but the average daily change of vehicles is very few, with an average of only 51.There are excessive bicycles deployed or stayed in the area, and the activity of more than 300 bicycles is not high.

(ii) Cluster B: $z\_Abn\_\mathrm{DAY} < 0$, $z\_cv < 0$: we call it a normal inefficient mode. Its $z\_cv < 0$ is as same as Cluster A, but $z\_Abn\_\mathrm{DAY} < 0$ compared with A indicating that the number of bicycles in the cluster area in this group is less than A. There are about 185 bicycles with an average of 17 bikes daily used, and more than 150 are not very active.

(iii) Cluster C: $z\_Abn\_\mathrm{DAY} < 0$, $z\_cv > 2$: this cluster has the highest *cv* value among the four clusters, indicating that the number of available bicycles in this group matches the demand for bicycles, and there are not too many idle bicycles available. The average number of available bicycles in this group is 213, and the average daily change is 117. More than half of the bicycles are used, so it is called high efficient mode.

(iv) Cluster D: $z\_Abn\_\mathrm{DAY} < 0$, $z\_cv > 0$: this cluster is similar to C, except that the *z_cv* value is lower than that of the C class but exceeds the average. In this group, the average number of available bicycles is almost similar to C, but *Abn*_DAY is about half of that of C, and its average *cv* is 2-3 times that of groups A and B. The use efficiency of bicycles is higher than that of A and B but lower than C, so it is called normal effective mode.

In general, high inefficient mode of cluster A has the max average number of available bikes and high efficiency mode of cluster C has the largest average coefficient of variation. The difference between A and B is in the average number of available bikes, and the difference between B, C, and D is in the coefficient of variation. From the number of the clusters, cluster A and cluster B together account for about 73%, so

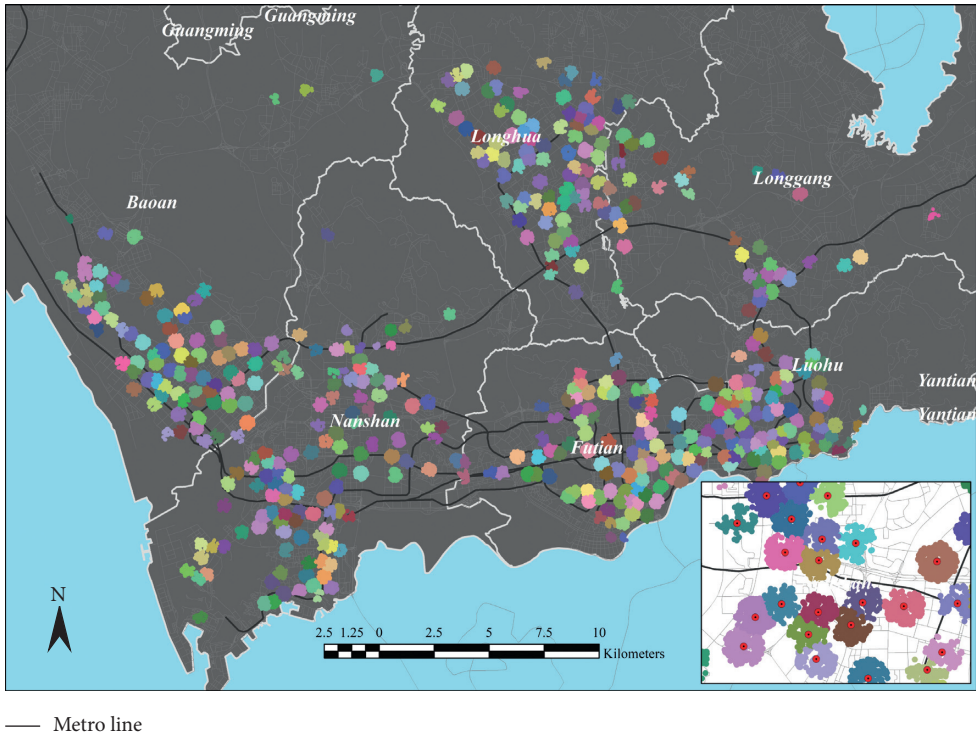


Figure 3: The OFO bicycle gathering area identified by mean shift clustering.

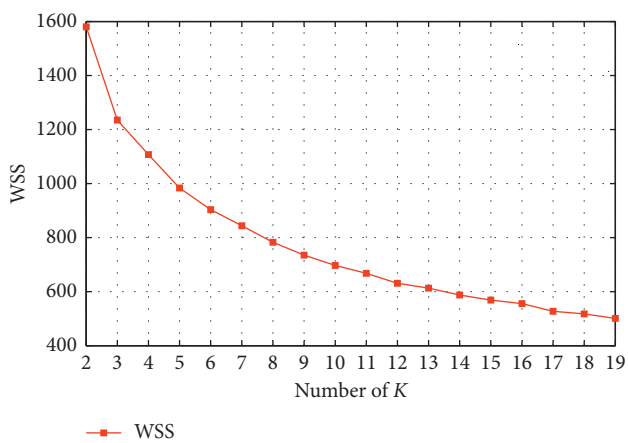


Figure 4: The WSS curve in determining the number of clusters.

the main mode is inefficient mode. These two groups have gathered a total of 110,000 bicycles which used low efficiency. The efficient mode area accounts for about 27% of all regions, of which the high efficient mode area accounts for 10%, with 10,000 bicycles, and the normal efficient mode accounts for 17%, totaling 17,705 bicycles. A total of 27,000 bicycles are gathered in these two groups, and the bicycle use efficiency in these areas is higher.

5.3.2. The Impact of Clustering of Bicycle Gathering Area. Figure 6 shows the spatial distribution of four clusters. The high inefficient mode is mainly distributed in the central area of Shenzhen (Futian and Luohu) and Baoan and shows the characteristics of spatial clustering. Normal inefficient areas are distributed on the periphery of urban built-up areas, and efficient mode areas are scattered between normal inefficient and high inefficient areas. It is worth noting that in Futian and Luohu districts where the subway network density is high, the main distribution mode is inefficient. In order to better understand the influencing factors that affect the types of bicycle service clusters, we analyzed the importance of the factors that determine the types of bicycle service clusters based on the RFC model with the highest accuracy.

The importance indicates how important the variables are in the RFC model. The sum of the importance of all variables is 1. Figure 7 shows the importance of 25 factors in the RFC model in ascending order when $K = 4$ and the average importance is 0.04 (1/25). The importance of population and buildings number is significantly higher than other factors which are key factors. The least important factors are the length of the branch road, the number of bus stops, and the area of public buildings. The importance of these three variables does not exceed 0.02 lower than the average. The importance of the main road length and the restaurant number ranked third and fourth, indicating that they are important reference variables for identifying bike gathering area type. The importance of resident building area, building coverage, distance to universities, and small shops is slightly larger than average. The distance to the subway station ranks only ninth in importance, which is about the same as the commercial building area and the company number. The sum of the importance of the top 10 variables accounted for 54%. Existing studies have shown that the area around subway stations is the most active area

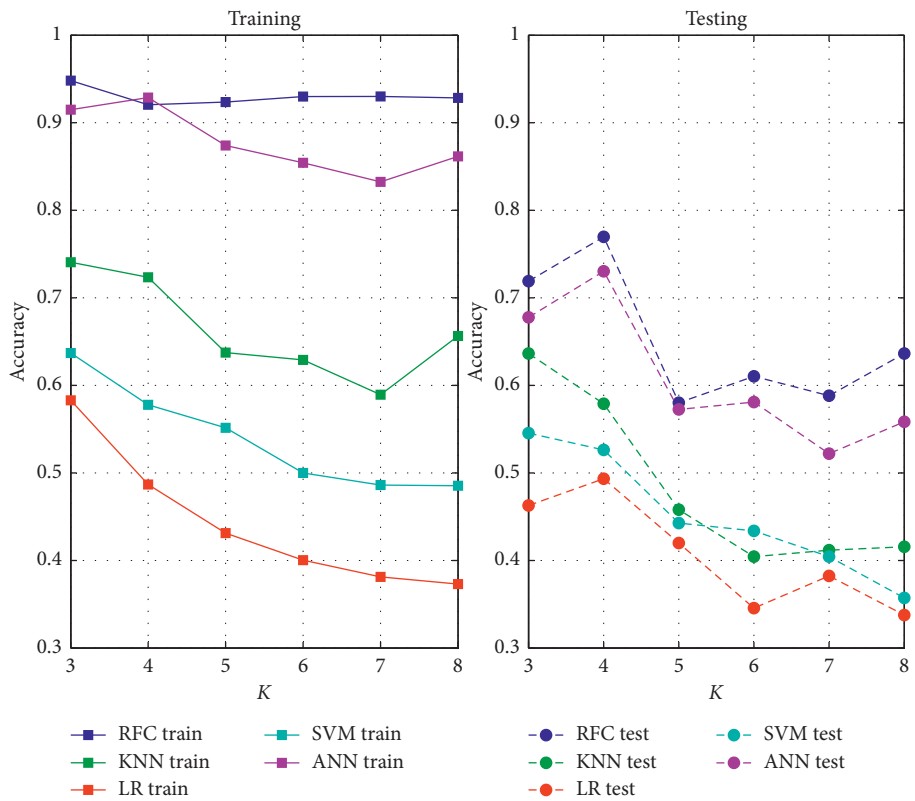


Figure 5: The accuracy of the five classification algorithms.

Table 2: The description of four clusters.

Cluster	A	B	C	D
Description	High inefficient	Normal inefficient	High efficient	Normal efficient
*z_ Abn*_DAY	1.03	−0.73	−0.52	−0.55
z_ cv	−0.39	−0.59	2.35	0.75
Abn_DAY	416	185	213	209
cv	0.12	0.09	0.55	0.30
Abn_DAY ∗ *cv*	51	17	117	63
Total available bike number	78588	31574	9986	17705
Cluster number	189	171	47	85

for bicycle activities, but our research has found that the distance from the subway station is not the most important factor in judging the activity type of bicycle gathering areas. The population, the number of buildings, the length of the main road, and restaurants are four important variables for judging the active types of bicycle clusters. In addition, among the 25 influencing factors in Figure 7, except for the variables with higher and lower importance, the importance of most of the variables in the middle is more evenly distributed, indicating that the active types of bicycle clusters have more and more complex influencing factors.

Figure 8 shows a comparison of the average values of the standard values of 25 factors. The color of the heat map clearly shows that there are obvious differences in the values of 25 variables between the four groups. We found that extreme values of variables tend to appear in groups A and C. Group A is significantly higher than the other three groups in the seven variables of population, number of buildings, length of main roads, number of restaurants, building coverage, number of parking lots, and number of bus stops. The number of population buildings and the length of secondary roads in group C are significantly lower than the other three groups, and the distance to school, industrial building area, office area, and distance to the park are significantly higher than the other three groups. Although the variables in groups B and D rarely have maximum or minimum values, the characteristics of the variables between them are quite different. The classification is based on average available number and the coefficient of variation, but the 25 variables between the classes

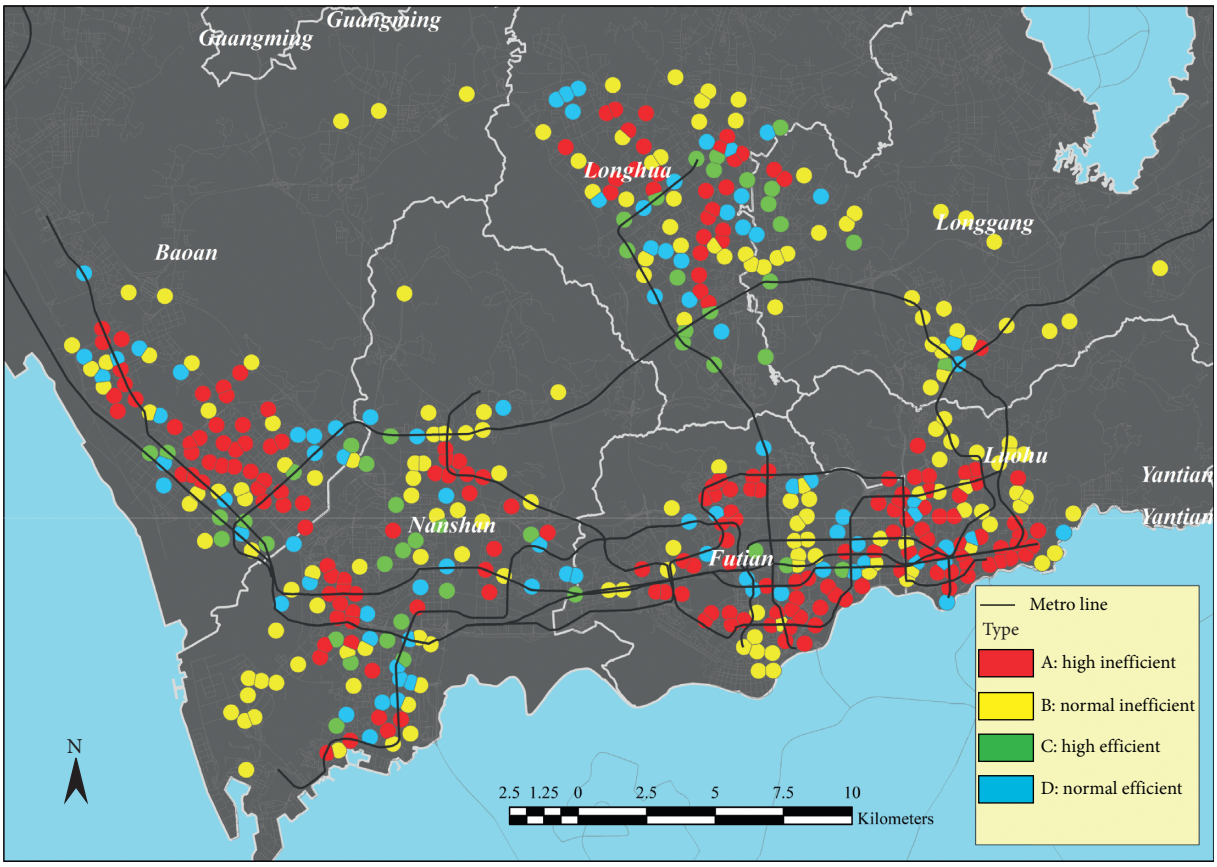


Figure 6: The distribution of four clusters.

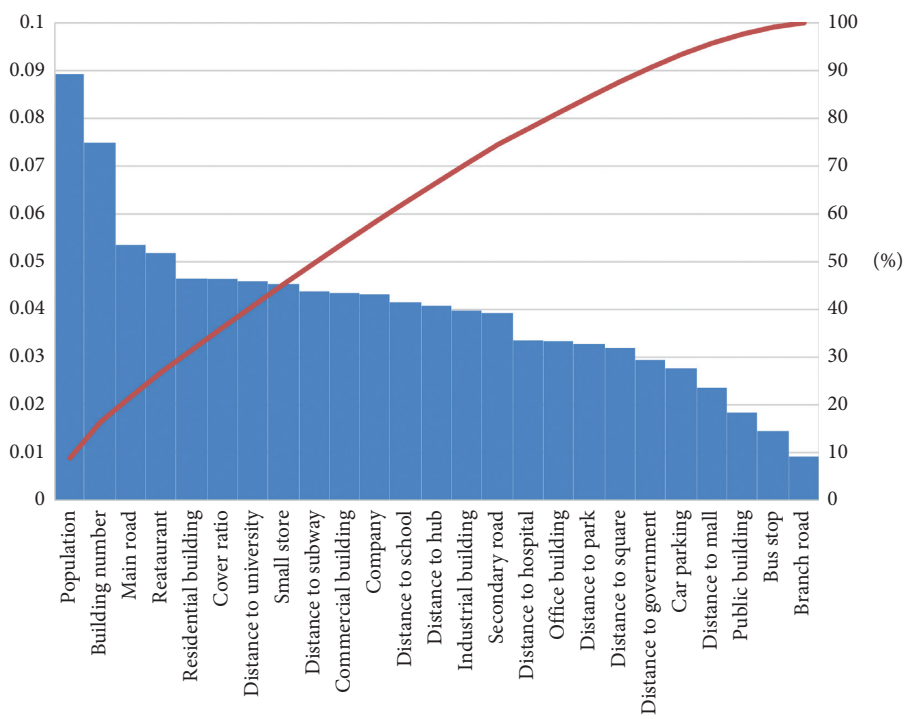


Figure 7: The importance of factor in random forest classifier model.

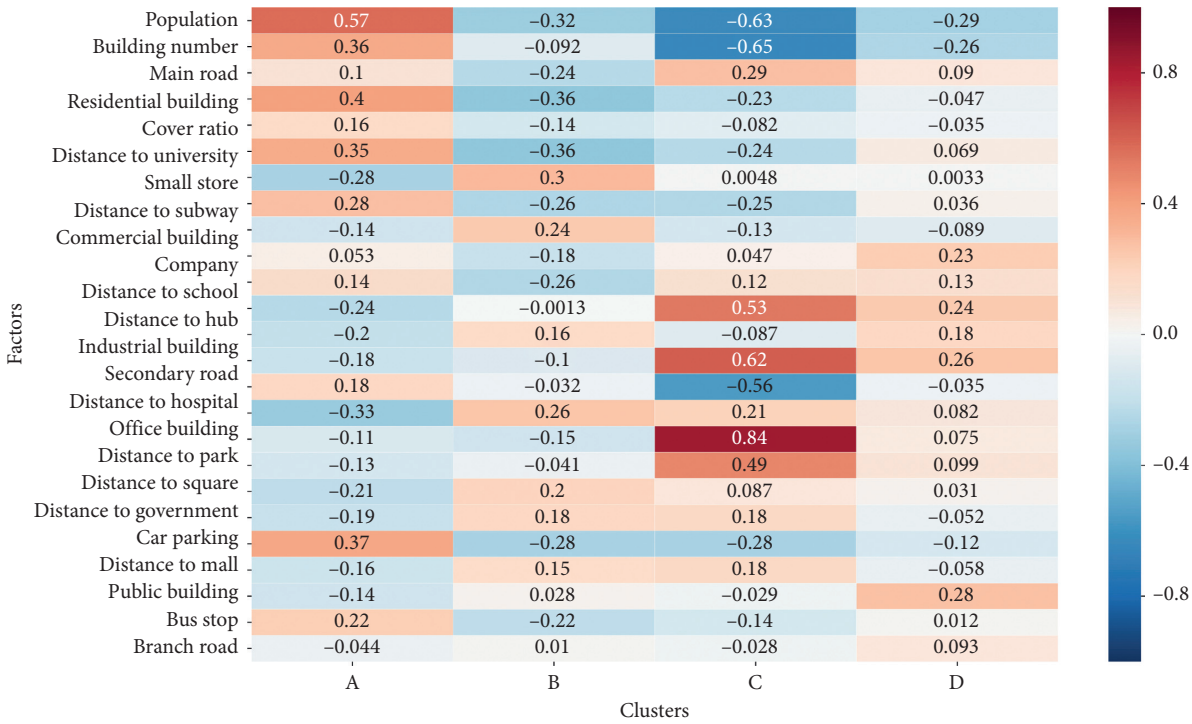


Figure 8: The heatmap of factor in four clusters.

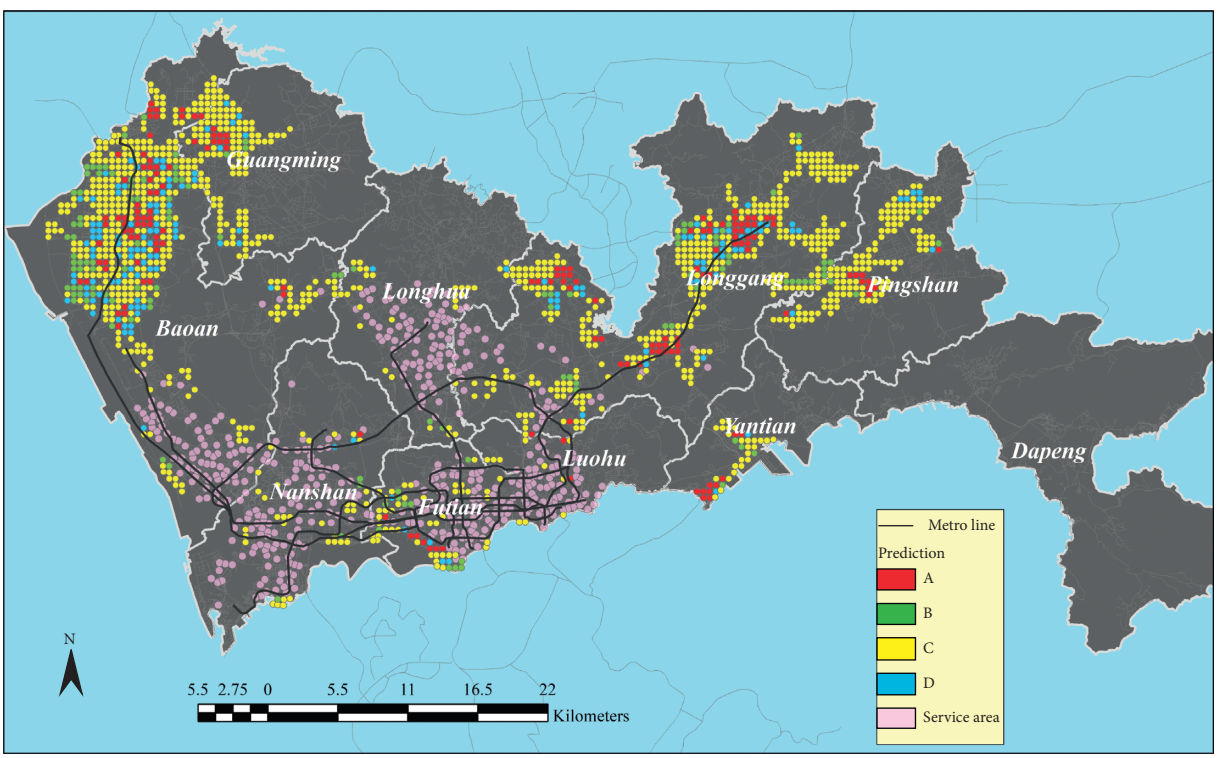


Figure 9: The predicted results of activity patterns in the new service area.

have obvious differences in value, indicating that the activity types of bicycles are related to these factors. The values of these variables can be used to judge the activity types of bicycles.

5.3.3. Guiding Public Bicycle Planning and Management. Let us assume a scenario: to provide a dockless public bicycle service in a new area in Shenzhen. Considering that the influencing variables are easy to obtain, we apply the RFC to

predict the activity pattern of the new service area. The built area of Shenzhen which does not provide OFO service is divided into 1459 grids, and the influence factors are calculated. Figure 9 shows the predicted results of the RFC model for activity patterns. Note that the prediction assumes that the existing bike operational and deployment strategies of OFO remain unchanged. Yellow grids belonging to cluster B have the largest number about 1025. There are 146 blue grids which are cluster D. The remaining area is 162 red grids and 126 green grids. Most girds are normal inefficient types. Table 2 and the service area type can provide information to public bicycle quantity management. We can get theoretically the minimum number of bicycles needed according to the grid's cluster and the total number of bicycles required. It is possible to optimize the number of available bicycles according to bike activity of the grid, reduce operating costs, and improve utilization efficiency.

6. Conclusions

This research practiced the introduction of a machine learning approach to quantity management using OFO bike operation data in Shenzhen. The contributions are mainly reflected in the following three aspects. First, we proposed a method for identifying the cluster area of dockless shared bicycles, which can accurately calculate the impact factors of shared bicycle systems. Second, different from previous research perspectives, this research discusses the performance and optimization possibilities of the shared bicycle system from the number of available bicycles in the gathering area and its changes. At last, this work shows the applicability and operability of machine learning methods in solving urban planning and management problems, which is inspiring for people with urban management background to use computational intelligence.

The bicycle gathering area type's recognition, prediction, and application in this study are meaningful for the sustainable development of shared bicycles. (1) There were 492 OFO bicycle gathering areas containing more than 140,000 bicycles, accounting for 63.6% of all bicycles in Shenzhen. (2) More type number of bicycle gathering area will affect the accuracy of the classification algorithm. The random forest classification had the best performance in identifying bicycle gathering area with an accuracy of more than 75%. (3) Shenzhen OFO dockless public bike gathering areas can be divided into four types: high inefficient, normal inefficient, high efficient, and normal efficient. The main area types are high inefficient and normal inefficient which gathered about 110,000 bicycles with low usage.(4) There were obvious differences in the characteristics of impact factors in four types of bicycle gathering areas. It is feasible to use these factors to predict area type to optimize the number of available bicycles, reduce operating costs, and improve utilization efficiency. So, the knowledge from the existing dockless bicycle operation data can be used to guide public bicycle planning and management. The potential activity patterns and the minimum number of bikes in new service areas can be obtained in advance. Operating companies can make optimization strategies based on this information.

Our study also has some limitations. First, due to the limitations of data acquisition, the working day operational data used only contain one week, so the results of the analysis may be biased. The data we analyzed did not include data for nonworking days. Weekend public bicycle usage patterns may differ from weekday. Second, the modes analyzed in this study rely heavily on operational data and may not be applicable elsewhere. When the strategy of bicycle operation changes or the number of bicycles is reoptimized, the activity modes will be affected. After a period of operation, according to the indicators and models of this study, a new mode of activity will be formed. Last, this paper focuses on the number of available bicycles, and the activity indicators of dockless PBS need to be further explored.

Data Availability

The bike data, population distribution data, and POI data including land use and built environment were supplied by the authors of this article. They are freely available. Requests for access to these data should be made to Qingfeng Zhou, zhouqingfeng@hit.edu.cn.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Financial support from the National Natural Science Foundation of China (grant no. 41771169) is acknowledged.

References

[1] P. Demaio, "Bike-sharing: history, impacts, models of provision, and future," *Journal of Public Transportation*, vol. 12, no. 4, pp. 41–56, 2009.

[2] China Academy of Information and Communications Technology, *China Shared Bicycle Industry Development Report*, China Academy of Information and Communications Technology, Beijing, China, 2018.

[3] Y. Zhou, B. P. L. Lau, Z. Koh, C. Yuen, and B. K. K. Ng, "Understanding crowd behaviors in a social event by passive WiFi sensing and data mining," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4442–4454, 2020.

[4] K. Li, C. Yuen, S. S. Kanhere et al., "An experimental study for tracking crowd in smart cities," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2966–2977, 2019.

[5] Q. Chen, W. Wang, F. Wu et al., "A survey on an emerging area: deep learning for smart city data," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 5, pp. 392–410, 2019.

[6] Y. Zhou, B. P. L. Lau, C. Yuen, B. Tuncer, and E. Wilhelm, "Understanding urban human mobility through crowdsensed data," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 52–59, 2018.

[7] L. P. L. Billy, N. Wijerathne, B. K. K. Ng et al., "Sensor fusion for public space utilization monitoring in a smart city," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 473–481, 2017.

[8] X. Wang, C. Yuen, N. U. Hassan et al., "Electric vehicle charging station placement for urban public bus systems,"

IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 1, pp. 128–139, 2017.

[9] A. Faghih-Imani and N. Eluru, "Analysing bicycle-sharing system user destination choice preferences: chicago's Divvy system," *Journal of Transport Geography*, vol. 44, pp. 53–64, 2015.

[10] C. Fricker and N. Gast, "Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity," *EURO Journal on Transportation and Logistics*, vol. 5, no. 3, pp. 261–291, 2016.

[11] P.-S. You, P.-J. Lee, and Y.-C. Hsieh, "An artificial intelligent approach to the bicycle repositioning problems," *Engineering Computations*, vol. 34, no. 1, pp. 145–163, 2017.

[12] E. O'Mahony and D. B. Shmoys, "Data analysis and optimization for (citi) bike sharing," in *Proceedings of the Twenty-ninth AAAI Conference on Artificial Intelligence*, AAAI Press, Austin, TX, USA, January 2015.

[13] Y. Chen, Y. Li, H. Hu, J. Zhang, D. Gu, and P. Xu, "Computational intelligence approaches to robotics, automation, and control," *Mathematical Problems in Engineering*, vol. 2015, Article ID 620275, 1 page, 2015.

[14] Áo Lozano, J. De Paz, G. Villarrubia González, D. Iglesia, and J. Bajo, "Multi-agent system for demand prediction and trip visualization in bike sharing systems," *Applied Sciences*, vol. 8, no. 1, p. 67, 2018.

[15] A. Faghih-Imani, N. Eluru, A. M. El-Geneidy, M. Rabbat, and U. Haq, "How land-use and urban form impact bicycle flows: evidence from the bicycle-sharing system (BIXI) in Montreal," *Journal of Transport Geography*, vol. 41, pp. 306–314, 2014.

[16] G. R. Krykewycz, C. M. Puchalsky, J. Rocks, B. Bonnette, and F. Jaskiewicz, "Defining a primary market and estimating demand for major bicycle-sharing program in philadelphia, Pennsylvania," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2143, no. 1, p. 117, 2010.

[17] W. El-Assi, M. Salah Mahmoud, and K. Nurul Habib, "Effects of built environment and weather on bike sharing demand: a station level analysis of commercial bike sharing in Toronto," *Transportation*, vol. 44, no. 3, pp. 589–613, 2017.

[18] R. C. Hampshire and L. Marla, *An Analysis of Bike Sharing Usage: Explaining Trip Generation and Attraction from Observed Demand*, Transportation Research Board, Washington, DC, USA, 2012.

[19] Y. Zhang, T. Thomas, M. Brussel, and M. van Maarseveen, "Exploring the impact of built environment factors on the use of public bikes at bike stations: case study in Zhongshan, China," *Journal of Transport Geography*, vol. 58, pp. 59–70, 2017.

[20] L. K. Maurer, *Feasibility Study for a Bicycle Sharing Program in Sacramento, California*, Transportation Research Board, Washington, DC, USA, 2011.

[21] K. Gebhart and R. B. Noland, "The impact of weather conditions on bikeshare trips in Washington, DC," *Transportation*, vol. 41, no. 6, pp. 1205–1225, 2014.

[22] D. Buck and R. Buehler, *Bike Lanes and Other Determinants of Capital Bikeshare Trips*, Transportation Research Board, Washington, DC, USA, 2012.

[23] X. L. G. S. Wang, "Modeling bike share station activity: effects of nearby businesses and jobs on trips to and from stations," *Journal of Urban Planning & Development*, vol. 142, no. 1, 2012.

[24] R. A. Rixey, "Station-level forecasting of bikesharing ridership," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2387, no. 1, pp. 46–55, 2013.

[25] J. T. Wong and C. Y. Cheng, "Exploring activity patterns of the Taipei public bike sharing system," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 11, pp. 1012–1028, 2015.

[26] J. Froehlich, J. Neumann, and N. Oliver, "Measuring the pulse of the city through shared bicycle programs," in *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems-UrbanSense 08*, Raleigh, NC, USA, November 2008.

[27] P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding bike-sharing systems using data mining: exploring activity patterns," *Procedia - Social and Behavioral Sciences*, vol. 20, pp. 514–523, 2011.

[28] N. Lathia, S. Ahmed, and L. Capra, "Measuring the impact of opening the London shared bicycle scheme to casual users," *Transportation Research Part C: Emerging Technologies*, vol. 22, no. 5, pp. 88–102, 2012.

[29] P. Borgnat, P. Abry, P. Flandrin, C. Robardet, J.-B. Rouquier, and E. Fleury, "Shared bicycles in a city: a signal processing and data analysis perspective," *Advances in Complex Systems*, vol. 14, no. 3, pp. 415–438, 2011.

[30] O. O Brien, J. Cheshire, and M. Batty, "Mining bicycle sharing data for generating insights into sustainable transport systems," *Journal of Transport Geography*, vol. 34, pp. 262–273, 2014.

[31] C. Etienne and O. Latifa, "Model-based count series clustering for bike sharing system usage mining: a case study with the vélib' system of paris," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, 2014.

[32] X. L. Zhou, "Understanding spatiotemporal patterns of biking behavior by analyzing massive bike-sharing data in Chicago," *PLoS One*, vol. 10, no. 10, 2015.

[33] L. Caggiani, R. Camporeale, M. Ottomanelli, and W. Y. Szeto, "A modeling framework for the dynamic management of free-floating bike-sharing systems," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 159–182, 2018.

[34] S. Reiss and K. Bogenberger, "GPS-data analysis of munich's free-floating bike sharing system and application of an operator-based relocation strategy," in *Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems-(ITSC 2015)*, September 2015.

[35] A. Pal, Yu Zhang, and C. Kwon, *Analyzing Mobility Patterns and Imbalance of Free-Floating Bike Sharing Systems*, Transportation Research Board, Washington, DC, USA, 2017.

[36] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Pearson Addison-Wesley, Boston, MA, USA, 2006.

[37] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[38] D. Comaniciu and P. Meer, "Mean shift: a Robust approach towards feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, 2002.

[39] M. Fernández-Delgado and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.

[40] L. Breiman, L. Breiman, and R. A. Cutler, "Random forests machine learning," *Journal of Clinical Microbiology*, vol. 2, pp. 199–228, 2001.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8861942, 15 pages
https://doi.org/10.1155/2020/8861942

Research Article

Generative Adversarial Network-based Missing Data Handling and Remaining Useful Life Estimation for Smart Train Control and Monitoring Systems

Hyunsoo Lee,[1] **Seok-Youn Han**,[2] **and Kee-Jun Park**[2]

[1]*School of Industrial Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea*
[2]*Urban Transit Research Group, Advanced Railroad Vehicle Division, Korea Railroad Research Institute, Uiwang, Republic of Korea*

Correspondence should be addressed to Hyunsoo Lee; hsl@kumoh.ac.kr

Received 29 July 2020; Revised 5 November 2020; Accepted 17 November 2020; Published 27 November 2020

Academic Editor: Ladislav Routil

As railway is considered one of the most significant transports, sudden malfunction of train components or delayed maintenance may considerably disrupt societal activities. To prevent this issue, various railway maintenance frameworks, from "periodic time-based and distance-based traditional maintenance frameworks" to "monitoring/conditional-based maintenance systems," have been proposed and developed. However, these maintenance frameworks depend on the current status and situations of trains and cars. To overcome these issues, several predictive frameworks have been proposed. This study proposes a new and effective remaining useful life (RUL) estimation framework using big data from a train control and monitoring system (TCMS). TCMS data is classified into two types: operation data and alarm data. Alarm or RUL information is extracted from the alarm data. Subsequently, a deep learning model achieves the mapping relationship between operation data and the extracted RUL. However, a number of TCMS data have missing values due to malfunction of embedded sensors and/or low life of monitoring modules. This issue is addressed in the proposed generative adversarial network (GAN) framework. Both deep neural network (DNN) models for a generator and a predictor estimate missing values and predict train fault, simultaneously. To prove the effectiveness of the proposed GAN-based predictive maintenance framework, TCMS data-based case studies and comparisons with other methods were carried out.

1. Introduction

Railway infrastructure has been one of the essential infrastructures not only at a national level, but also across continents. In terms of ground cargo and freight transport, the railway system is the most important infrastructure. A number of research studies have focused on detections of aberrant situations in trains. For instance, unexpected failures in train components may catastrophically harm the passengers' safety. Moreover, a maintenance delay may result in subsequent heavy delays in overall train schedule. Thus, maintenance frameworks for railway infrastructure have received significant attention. A number of existing research studies have proposed various railway maintenance frameworks and relevant applications for more reliable railroad operations.

Early railway maintenance frameworks are based on periodic time-based maintenance [1], which is still an effective technique for checking railway components. Monthly or quarterly inspection belongs to this type of maintenance. Recently, maintenance framework has evolved to "preventive maintenance" in contemporary railroad systems. Preventive maintenance is classified into "time-based maintenance" and "distance-based maintenance" in general. Most railway operators utilize both maintenance frameworks, simultaneously.

Moreover, the maintenance framework is evolving with the development of technologies of the fourth industrial revolutions. Of these technologies, Internet of Things (IoT) technology is the most relevant for enhancing railway maintenance. Fraga-Lamas et al. [2] summarized the

utilization of IoT technologies in train maintenance. IoT-based embedded systems enable the detection of abnormal status of railway components in real time, where the signals are subsequently transferred to a secured database system. In general, most of the train systems have their own management systems, such as train control and monitoring system (TCMS) for storing various train data and for managing trains. Based on TCMS-based research studies [3, 4], TCMS is a system with control, communication, and management functions for all train platforms and applications. As the system collects operation and management-based data for trains and their connected cars, a huge amount of data can be collected and analysed. Several research studies [5, 6] used TCMS data for estimating energy consumption of trains or for controlling train doors safely. However, relatively fewer studies on predictive maintenance were carried out.

This study focuses on developing a new and effective predictive maintenance framework using TCMS data. In this study, remaining useful lives (RUL) of various train modules are predicted using a proposed deep learning method. In order to measure RUL of train modules, this study predicts time periods to the relevant trains' faults and malfunctions. In this paper, this time to failure (TTF) for a certain train fault is defined a RUL of a certain fault. However, prerequisite conditions are necessary to handle data issues in TCMS. As TCMS data could include missing values, their handling mechanism must be embedded in a relevant RUL estimation framework.

This study applies a generative adversarial network (GAN) to handle missing values in TCMS. The following section provides relevant background knowledge and literature review. Section 3 examines TCMS data and relevant data issues. Sections 4 and 5 present a GAN-based predictive maintenance framework and its verifications using various numerical analyses, respectively.

2. Background and Literature Review

This study utilized TCMS data to estimate train component status and predict their RULs. The proposed framework is classified as a predictive maintenance framework in train systems. As discussed in the previous section, the maintenance paradigm in train transportation has converged with the technologies of the fourth industrial revolution. The time and distance-based maintenance frameworks have been combined with monitoring-based methods. Several sensing systems have been developed and installed for more detailed examinations of trains' components. Sharma et al. [7] detected breakage of railway tracks using vibration sensors. Sireesha et al. [8] used a radio frequency-based method to detect rails' broken status. These sensing systems have integrated with Internet of Things (IoT-) based frameworks. Lee [9, 10] developed various Industrial IoT (IIoT) systems to monitor abnormal manufacturing signals and estimate production performance indices in multiple supply chains. The detected signals are transmitted to a cloud server, where industrial big data analytics analyses the collected data and takes preventive measures for better production controls. These technologies and frameworks have been applied to various train systems and their relevant monitoring-based/condition-based maintenance. Hitachi [11] proposed Lumada IoT Platform© as a monitoring-based maintenance system for its railway system.

While various monitoring-based methods for detecting abnormal status of railway components have been introduced, deep learning methods and relevant data analytics have been integrated into predictive maintenance. Corman et al. [12] applied a data-driven method to estimate the remaining life of a light rail braking system in a train. McKinsey [13] suggested similar approaches to enhance rail operations using digital maintenance technologies. Atamuradov et al. [17] and Liden [18] summarized comprehensive overviews on railway infrastructure maintenance. Table 1 provides various time-based, monitoring-based, and data-driven maintenance frameworks and their applications.

As shown in Table 1, data driven analytics has been introduced for better rail maintenance. Among a number of data sets in a train framework, the TCMS data is the most comprehensive data, as it includes operation, parameter settings, and other information on train components. Figure 1 shows the various TCMS subcomponents that are installed in Korean trains and cars.

In general, TCMS is an essential system for controlling electrical multiple units (EMU) in each train and car in a train system. Thus, control parameters and operation data are stored in TCMS. While TCMS is mainly used to control trains and cars, the usage of TCMS data for various purposes has been suggested. Table 2 shows various applications that use TCMS data. As shown in Table 2, most of the applications that use TCMS data have focused on monitoring-based maintenance.

While TCMS data have been used comparatively less with more advanced maintenance analytics, several industrial projects including Shift2Rail [21] have suggested predictive maintenance frameworks using TCMS data. However, these projects provide only conceptual frameworks or experimental-level demonstrations. In particular, big data analytics and more advanced data mining methods are seldom applied in TCMS-based predictive maintenance frameworks. To address this issue, this study proposes a new and effective predictive maintenance using deep learning methods and real-time TCMS data handling modules.

3. Missing Value Issues in TCMS Data for Predictive Maintenance Framework

The proposed predictive maintenance framework uses TCMS data for predicting RULs in a certain breakage. As shown in (1), the RUL (RUL_j, $j \in J$; J is a set of integers) of a certain breakage (j) is estimated using the TCMS data (X) and used as a main reference for setting up train and cars maintenance schedules. The function $f(\cdot)$ is modelled using a deep learning-based network architecture and is explained in the following section.

$$\mathrm{RUL}_j = f\left(x_{i \in I}\right). \tag{1}$$

Table 1: Existing time, monitoring, and data-driven maintenance applications in rail systems.

Existing research studies	Target railway components	Methods and characteristics	Maintenance type		
			Time/ distance	Monitoring	Data-driven
Faiz and Singh [14]	Railway track	(i) Detection of track geometry (ii) Usage of rail profile-based regression model	O	—	—
Sharma et al. [7]	Railway	(i) Vibration sensor-based estimation of railway breakage	—	O	—
Shaikh et al. [15]	Solid axle wheel sets	(i) Installation of additional sensors (vibration sensors for capturing lateral and yaw dynamics) (ii) Vibration model-based simulation	—	O	—
Letot et al. [16]	Railway track point machine	(i) Degradation assessment and data-based RUL estimation	—	—	O
Corman et al. [12]	Train breaking system	(i) Work, maintenance, and failure data-based reliability estimation (ii) Usage of Weibull distribution	O	O	O

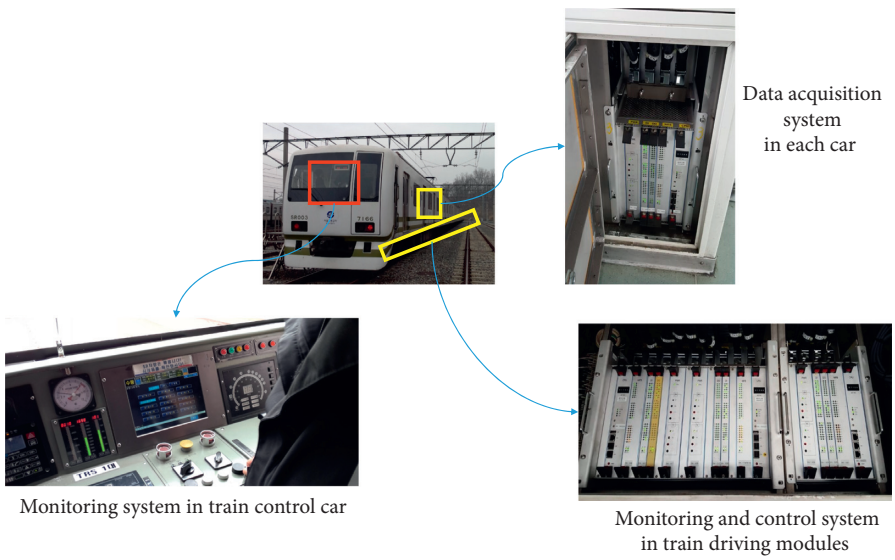


Figure 1: Train control and monitoring system (TCMS) in Korean trains and cars. (a) Monitoring system in train control car. (b) Monitoring and control system in train driving modules. (c) Data acquisition system in each car.

Table 3 summaries the TCMS data used in this study and the general specifications of the proposed RUL prediction framework.

As shown in Table 3, the input data of the proposed predictive maintenance framework is the TCMS data. Figure 2(a) shows a part of the TCMS data, which is an encrypted data. In general, TCMS data is classified into two types: operation data (oper) and alarm data (arm) as shown in Figure 2(a). The TCMS data is stored with each car no., the date, and time. While the "oper" data describes information such as train identification and operation and other relevant train parameters, the recorded "arm" data includes various warning signals and relevant alarm codes. These alarm level information and other warning data are written using the predefined criteria, such as status levels of train components and other relevant sensor measuring ranges. Thus, "oper" data is used as input data, while the output of the proposed RUL estimation framework is driven by the "arm" data. If "arm" data can be predicted using a series of "oper" data, a real-time predictive maintenance can be applied. Hence, the proposed predictive maintenance framework uses "oper" data as input vector. The RUL variable is extracted from the "arm" data and fault/maintenance history. The fault/maintenance data clarifies the relationship among "arm" data and a certain train defect. RUL data is extracted from the "arm" data and its relationship to a certain defect is obtained using the mapping between both data.

However, TCMS data cannot be directly used owing to their encrypted formats and missing value issue. As shown in Figure 2(a), both types of data are encrypted for various reasons, such as data protection, data size reduction, and sensor driver encryption. This indicates that the data need to be decrypted prior to any further data processing.

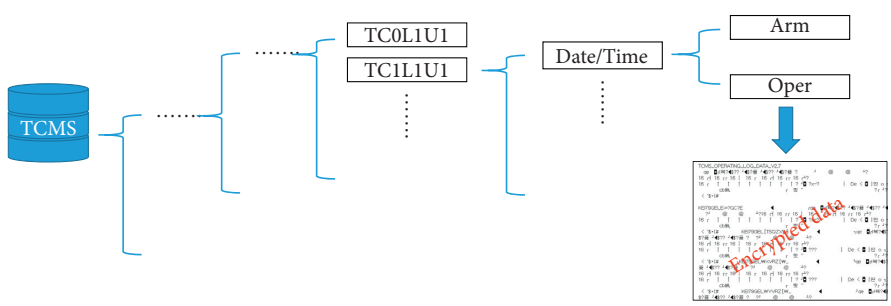
To decrypt the data, hex data-based decoding is performed as an essential prerequisite procedure using the encryption rule for TCMS. Then, the hex-formatted data are converted into number-formatted data for subsequent deep learning processing. Figure 2(b) shows a program developed

Table 2: Applications that use TCMS data.

Research studies	Applications	TCMS data
Ito et al. [6]	(i) Safe door operation (ii) Automatic power changer-based driver advisory system	(i) EMU functions in TCMS
Neil [19]	(iii) Railway safety monitoring-based maintenance	(ii) Transaction data in TCMS
Kim et al. [13]	(iv) Analysis of train energy consumption considering driving patterns	(iii) Driving time (iv) Train's driving speed (v) Railway track data
Xu et al. [20]	(v) Queuing theory-based maintenance cycle scheduling for an urban rail transit system	(vi) Running distance, velocity, and mileage data Maintenance schedule
Shift2Rail project [21]	(vi) Monitoring of cargo condition (vii) A conceptual and experimental project	(vii) TCMS data (viii) Additional sensing systems (e.g., ultrasonic sensor and other wireless sensors)

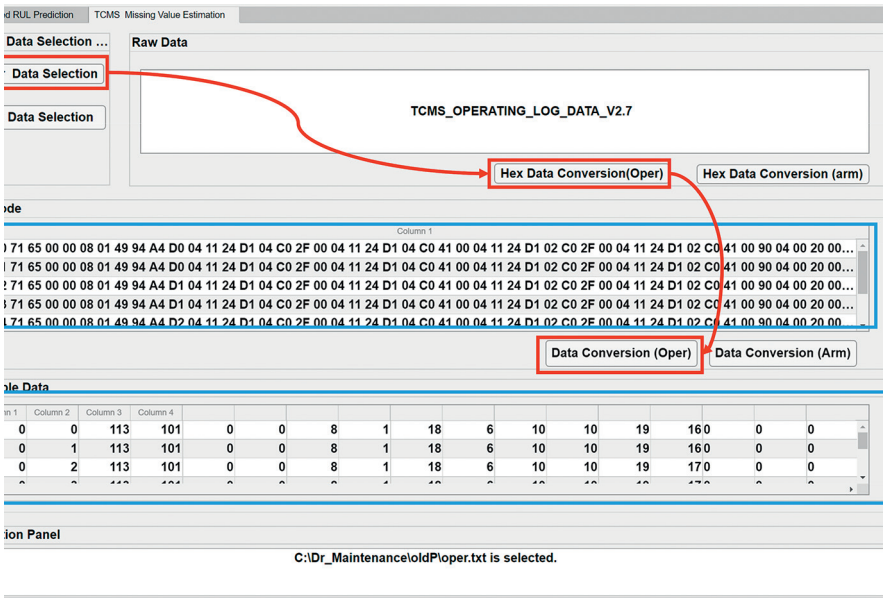
Table 3: Specification of the proposed predictive maintenance framework.

Content	Classification	Issues
Data specification	(i) Data source: TCMS data (2018.6~2019.05) (ii) Data from the seventh line in subway system, Republic of Korea	Big data
TCMS data specification	(i) Number of attributes: 2643 per one record Existence of a number of missing values in one record (ii) Data format: encrypted data	(i) Data decryption is needed (ii) Missing value handling is needed
Fault/alarm data	(i) Number of attributes: 56 (ii) Data format: encrypted text data	(i) Data decryption is needed
Predictive maintenance framework	(i) Data input: TCMS data (ii) Output: the estimated RUL (iii) Mechanism: GAN-based deep neural network	—



(a)

Figure 2: Continued.



(b)

Figure 2: Encrypted TCMS data sample and the developed conversion tool. (a) A TCMS data sample and its encrypted texts. (b) The decryption and conversion software program developed in this study.

in this study, which converts hex-formatted data to number-formatted interpretable data. The "oper" data has 2,643 attributes, which include train identification, operation time, station time, velocity, sensor data, and other information. The "arm" data has 56 attributes such as system failure, warning, and alarm information.

The data conversion process is performed as a pre-processing step. However, the main issue is the frequent occurrence of missing values in the TCMS data. Missing values in the TCMS data exist due to various reasons (e.g., sudden breakdown of sensors, malfunction of devices, and/or sudden changes of electric current). Missing input and output data issues have to be resolved prior to training a deep learning-based predictive maintenance model. This is one of the most common issues in manufacturing [22–24], transportation, and other data handling processes. Table 4 summaries various methods for handling missing values.

As shown in Table 4, most of the early relevant research studies tried to remove records with missing values. While these methods provided complete data for input, it could result in the lack of a training data. However, this limitation can be overcome by estimating missing values. The simplest estimation method is to consider data in the same attribute and then extract a probability density function using the data associated with the same attribute. For instance, the Gaussian mixture model (GMM) can be applied for capturing the characteristics of the data [30]. Then, a random number generated using the reasoned GMM model is used as an estimator for a missing value. However, in this method, relationships among other attributes are ignored. To address this issue, another estimation method, missing value estimation method, which considers the overall dependencies among the data attributes can be used. In general, multivariate statistical approaches, regression model, or multivariate nearest neighbour methods [31] can be applied for describing data dependencies. Then, the missing values can be estimated and substituted using Markov Chain Monte Carlo (MCMC)-based random number generation methods.

While these methods have worked only for estimating missing values, the recent methods tend to generate not only missing data, but also overall data. The generative adversarial network (GAN) is the representative method among them. In industrial big data, one of the issues is the lack of certain fault data. The lack of certain types of data may lower training performances of applied learning mechanisms. While the initial purpose of adversarial network (G) in GAN is to increase the classification ability of a classification network (D), a well-trained adversarial network can generate data which fits to an objective. Table 5 summarizes the learning algorithm of GAN.

The gradients for G and D are driven using

$$\min_{G} \max_{D} f_D(X') + f_G(Z). \tag{2}$$

As shown in (2), $f_D(X')$ and $f_G(Z)$ denote $E_{X'}[\log D(X)]$ and $E_Z[\log(1 - D(G(Z)))]$, respectively. Several research studies applied GAN for generating fault data in automotive [22], semiconductor [23], and steel production processes [24]. This study used GAN to handle the missing value issues in the TCMS data as well as to predict RULs in train components.

4. Generative Adversarial Network-Based Predictive Maintenance Framework

The proposed RUL estimation framework predicts the RUL of a certain defect or a malfunction. To focus on major

Table 4: Various methods for handling missing values.

Methods for handling missing values	Detailed methods	Related research studies
Removals of data sets with missing values	(i) Ignorance of records with missing values (ii) Data without missing values are used only for an input vector	A number of research studies including [25]
Estimation of missing values (I)	(iii) Estimation of missing values using mean, MCMC, and nearest neighbours (iv) Estimation considering only the attribute that has missing values	Moldovan et al. [26]
Estimation of missing values (II), multiple imputation	(v) Data estimation considering overall attributes' dependency (vi) Missing values estimation using regression and other statistical methods	Hruschka et al. [27] Yuan [28]
Generation of a new data set	(vii) Generative adversarial network- (GAN-) based data generation (viii) Replacement of the data having missing values with newly generated data	Kim and Lee [23, 24] Douzas and Bacao [29]

Table 5: General learning algorithm of GAN.

Input/parameters	(i) Training data: X (ii) Learning epoch: $k1$/· Training epoch: $k2$ (iii) Step length: η (iv) Mini-batch size: m
Output	(v) Optimal parameters for G: $\widehat{\theta_G}$ (vi) Optimal parameters for D: $\widehat{\theta_D}$
Learning algorithm	for 1:$k1$ Initialize θ_G, θ_D for 1:$k2$ mini-batch partitioning from X, $X' = \{x_1, \ldots, x_m\}$ calculate gradient for D and update θ_D $\theta_D' = \theta_D + \eta \cdot (\partial f_D / \partial \theta_D)$ Generate random vector, $Z' = \{z_1, \ldots, z_m\}$ Calculate gradient for G and update θ_G $\theta_G' = \theta_G + \eta \cdot (\partial f_G / \partial \theta_G)$ end end

malfunctions during train operations, 49 defects are extracted from the "arm" data in TCMS based on defect frequency and severity. Figure 3 shows the defect frequency. The records are gathered by Korean Railroad Research Institute.

Each defect's RUL is calculated using the TCMS "arm" data and relevant fault/maintenance data. The "arm" data includes the identification number, occurrence date, and other relevant information for each defect. Figure 4 shows an occurrence history of a specific defect (defect code no. 442–fault of electronic control unit (ECU)). As shown in Figure 4, the X and the Y axes indicate the occurrence date and defect code, respectively.

From the information, $\text{Fault}_{i,j}(t)$ is extracted. $\text{Fault}_{i,j}(t)$ indicates the jth occurrence time of the ith defect in the TCMS data. Then, inter-defect time, $\text{RUL}_{i,j}(t)$, is calculated using

$$\text{RUL}_{i,j}(t) = \left(\text{Fault}_{i,j}(t) - \text{Fault}_{i,j-1}(t)\right). \tag{3}$$

As shown in (3), $\text{Fault}_{i,j}(t)$ denotes the jth sensing time of the ith defect in TCMS, and $\text{RUL}_{i,j}(t)$ indicates the inter-defect time in day between the jth occurrence and $(j-1)$th occurrence of the ith defect. The obtained RUL is used as output data for prediction. Subsequently, the RUL is predicted using TCMS's operation data ($X(t)$) and a deep learning framework as shown in (4).

$$\text{RUL}_{i,j}(t) = f_n\left(w_n \cdot \left(f_{n-1}\left(\ldots f_1\left(\sum_{i,j} w_i \cdot x_i(t) + b_i\right)\right)\right) + b_n\right), \tag{4}$$

$$X(t) = \left(x_1(t), \ldots, x_i(t), \ldots, x_{2643}\right). \tag{5}$$

As denoted in (4) and (5), $x_i(t)$ is the value of the ith attribute at time t in the TCMS "oper" data, w_i is the weight value of $x_i(t)$, b_i is the ith bias, and f_i is the ith activation function.

While a general predictive maintenance estimates RUL using (4), the equation cannot be directly applied in the TCMS-based data mining owing to the missing value issue discussed in the previous section. To overcome this issue, (4) is converted by

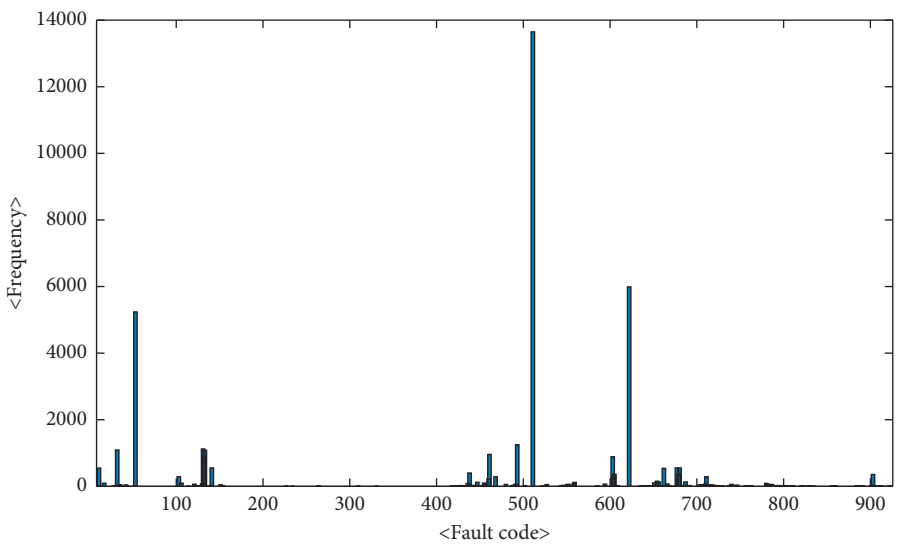


Figure 3: Defect frequency (each fault code is designated by Korean Railroad Research Institute).

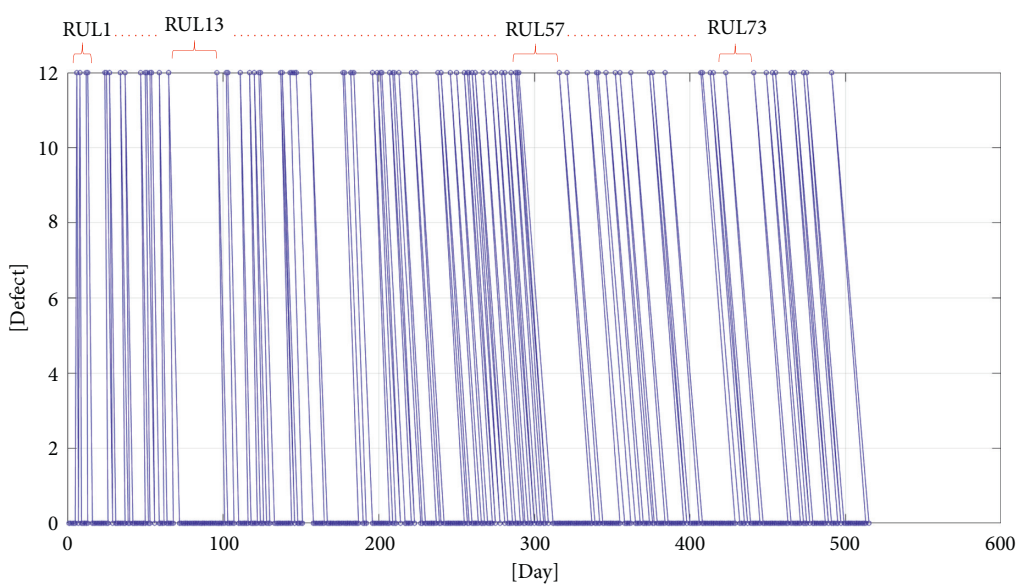


Figure 4: RUL extraction for a specific defect (code no. 442, fault of ECU) from the TCMS "arm" data.

introducing a GAN in the RUL estimation. Figure 5 shows the overall RUL prediction framework using GAN.

As shown in Figure 5, the proposed RUL prediction framework consists of two phases: learning stage and prediction stage. The main objective of the first stage is to obtain a discriminator (D) using a deep learning-based architecture. The proposed framework applies a deep neural network for the discriminator. The discriminator uses a complete TCMS data ($X'(t)$), where $X'(t)$ is generated using $X(t)$ and a generator (G) in the proposed GAN. $X'(t)$ is complete data, while $X(t)$ is a data set with missing values. As discussed in the previous section, the TCMS data ($X(t)$) of a certain train's fault may have missing values owing to various reasons. As shown in (6), these missing values are estimated initially using multivariate GMM, $p(\theta|x_i)$ where θ is the extracted RUL data, x_i is the ith attribute's data over the entire time considering $x_i(t)$, and $|\text{oper}|$ is the data size of x_i.

$$p(\theta|x_i) = \sum_{i=1}^{k} \phi_i \cdot N(\mu_i, \Sigma_i), \tag{6}$$

where $N(\mu_i, \Sigma_i) = e^{-(1/2)\cdot(x_i - mu_i)^T \cdot \Sigma^{-1} \cdot (x_i - mu_i)} / \sqrt{2\pi^{|oper|} \cdot |\Sigma_i|}$.

The missing value is generated using Gibbs sampling method [30, 31]. The initial completed data ($X'(t)$) is inputted to a generator $G(\cdot)$. The output of G is the regenerated data ($X''(t)$). The generator has another deep neural network architecture similar to discriminator $D(\cdot)$ as shown in (2). Then, D generates $X''(t)$ that satisfies (1) better

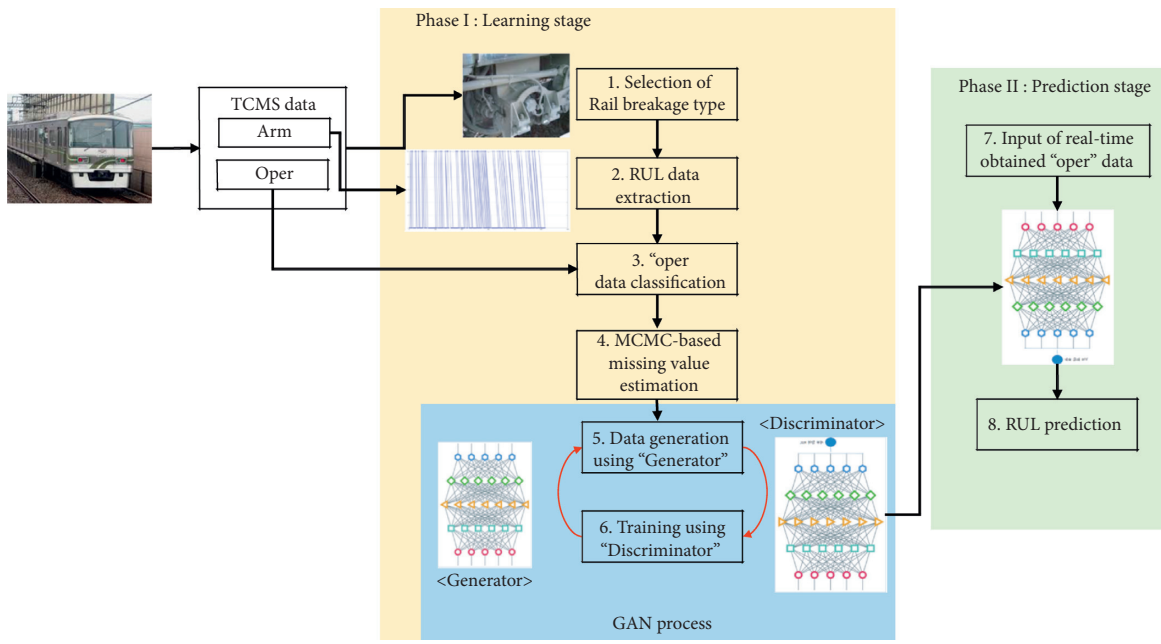


Figure 5: GAN-based missing value handling and RUL prediction framework.

than the previous estimated $X'(t)$. The updated $X''(t)$ is then inputted to $D(X''(t))$.

As presented in Figure 5, the GAN process shows these learning processes. Both network models have the deep learning parameters $\theta^{(D)}$ and $\theta^{(G)}$ as weight vectors. The updating procedures $\theta^{(D)}$ are achieved using the gradients indicated in (7)–(9).

$$\frac{\partial V}{\partial \theta^{(D)}} = \frac{\partial f_D(X''(t))}{\partial \theta^{(D)}} + \frac{\partial f_G(X'(t))}{\partial \theta^{(D)}}. \tag{7}$$

As shown in (7), V denotes $f_D(X''(t)) + f_G(X'(t))$.

$$\frac{\partial f_D(X''(t))}{\partial \theta^{(D)}} = \frac{\partial f_D(X''(t))}{\partial y_i^D} \cdot \frac{\partial y_i^D}{\partial v_i^D} \cdot \frac{\partial v_i^D}{\partial y_{i-1}^D}, \ldots, \varphi(v_1) \cdot X''(t). \tag{8}$$

v_i denotes $(w_i \cdot y_{i-1}) + b_i$, and y_i is the output of the ith deep learning layer, $\phi(v_i)$.

$$\frac{\partial f_G(X'(t))}{\partial \theta^{(D)}} = \frac{\partial f_G(X'(t))}{\partial y_i^D} \cdot \frac{\partial y_i^D}{\partial v_i^D} \cdot \frac{\partial v_i^D}{\partial y_{i-1}^D}, \ldots, \cdot \varphi(v_1) \cdot X'(t). \tag{9}$$

$\theta^{(G)}$ is updated using the same procedures. Finally, the learned $D(\cdot)$ is obtained after the overall learning iterations.

In the second phase (prediction stage), the RUL of a certain defect is estimated using real-time "oper" data. While real-time data $(X(t))$ may have missing values, the RUL is estimated successfully with $D(X''(t))$. The proposed GAN-based RUL prediction framework considers data with various missing values that exist frequently in TCMS data. While TCMS may generate data with missing values due to various issues, the proposed framework is considered an effective predictive maintenance framework for handling missing values. The following section proves the effectiveness of the proposed framework using case studies and TCMS data analyses.

5. Verification and Analysis of GAN-Based RUL Prediction Framework

To prove the effectiveness of the proposed TCMS-based predictive maintenance framework, this section provides prediction performances of several train faults and compares prediction accuracies with other methods. As explained in the previous section, each fault type's RUL was estimated using its GAN-based framework.

The prediction accuracy and analyses were performed using the data of the Korean Railroad Research Institute (KRRI), which is a Korean government-funded railroad institute. The TCMS data from June, 2018, to May, 2019, in Seoul Metropolitan Subway were used as training and testing data. Sixteen defects were selected to predict their RULs. Table 6 provides the fault types and their information. The fault code ID and other relevant information were recorded in the TCMS data of the Seoul Metropolitan Subway.

To prove the verification using the provided GAN-based missing value estimation, the proposed method was compared with the other three existing methods: (1) ARIMA-based RUL estimation, (2) estimation with pruning of missing values, and (3) RUL prediction using mean-value estimation. Table 7 summaries the architecture, characteristics, and parameters of the proposed method and the three existing methods.

The GAN architecture and other relevant parameters of the proposed method are provided in Table 8. As mentioned

Table 6: Fault types and their information for RUL predictions.

No.	Fault code ID	Fault information and relevant location
1	31	LIU1 communication error in TC1
2	32	LIU2 communication error in TC1
3	34	LIU2 communication error in TC0
4	38	LIU1 hardware malfunction in TC0
5	39	LIU2 hardware malfunction in TC
6	231	SIV inverter malfunction
7	434	Break malfunction
8	442	ECU malfunction
9	635	TC MFB card/ATC vital malfunction
10	636	Tachometer error
11	640	Main ATC hardware malfunction
12	641	Secondary ATC hardware malfunction
13	647	FSB/ATC error
14	669	ATO-ATC communication error
15	670	ATO-ATC 1 communication error
16	684	ATC DBAU hardware error

Table 7: Four RUL prediction methods.

RUL prediction method	GAN-based RUL estimation (the proposed method)	ARIMA-based RUL estimation	RUL estimation using "missing value pruning"	RUL estimation with "mean-value estimation" of missing values
Missing value handling mechanism	*O* (GAN-based data generation)	*X* (removal of records with missing values)	*X* (removal of records with missing values)	O (mean-value estimation)
RUL estimation method	Classification using GAN	ARIMA-based RUL estimation	Deep neural network	Deep neural network
Detailed parameters	Refer to Table 8	ARIMA (6, 2, 5)	(i) Learning epoch: 1000 (ii) Learning rate: 10^{-3} (iii) Number of layers:10 (iv) Used activation functions =(leaky RU for final layer, Sigmoid for layers #1–#9)	—

Table 8: Detailed architecture and relevant parameters of the GAN-based RUL estimation (case for fault code ID 31).

Classification	Detailed architectures
General learning parameters	(i) Learning epoch: 1000 (ii) Learning rate: 10^{-3}
Discriminator $(D(\cdot))$	(i) Number of Layers:10 (ii) Number of nodes in each layer =(1, 50, 100, 200, 500, 1500, 2500, 3000, 3500, 2653) (iii) Used activation functions =(leaky RU for final layer, Sigmoid for layers #1–#9)
Generator $(G(\cdot))$	(i) Number of layers: 7 (ii) Number of nodes in each layer =(2653, 2700, 2800, 3000, 3200, 3500, 2653) (iii) Used activation functions: Sigmoid function for each layer

in the previous section, the GAN architecture varies for every defect. The parameters are provided in Table 8.

The parameters shown in Tables 7 and 8 were determined by applying numerical tests on the provided TCMS data. Figure 6 shows the training and test accuracies of the proposed RUL prediction for fault code ID 31 (LIU1 communication error in TC1).

As shown in Figure 6, the proposed method performed 99.9% and 83.5% for the training and test accuracies, respectively. The accuracy was calculated using (10) and (11). The root mean squared error $(\mathrm{RMSE}(\mathrm{RUL})_i)$ for a certain type (i) of RUL (RUL_i) was used as a test metric, where $\widehat{\mathrm{RUL}}_{i,j}$ is the jth predicted value using the proposed framework and $\mathrm{RUL}_{i,j}$ is the jth original RUL value, and n is the size of the test data.

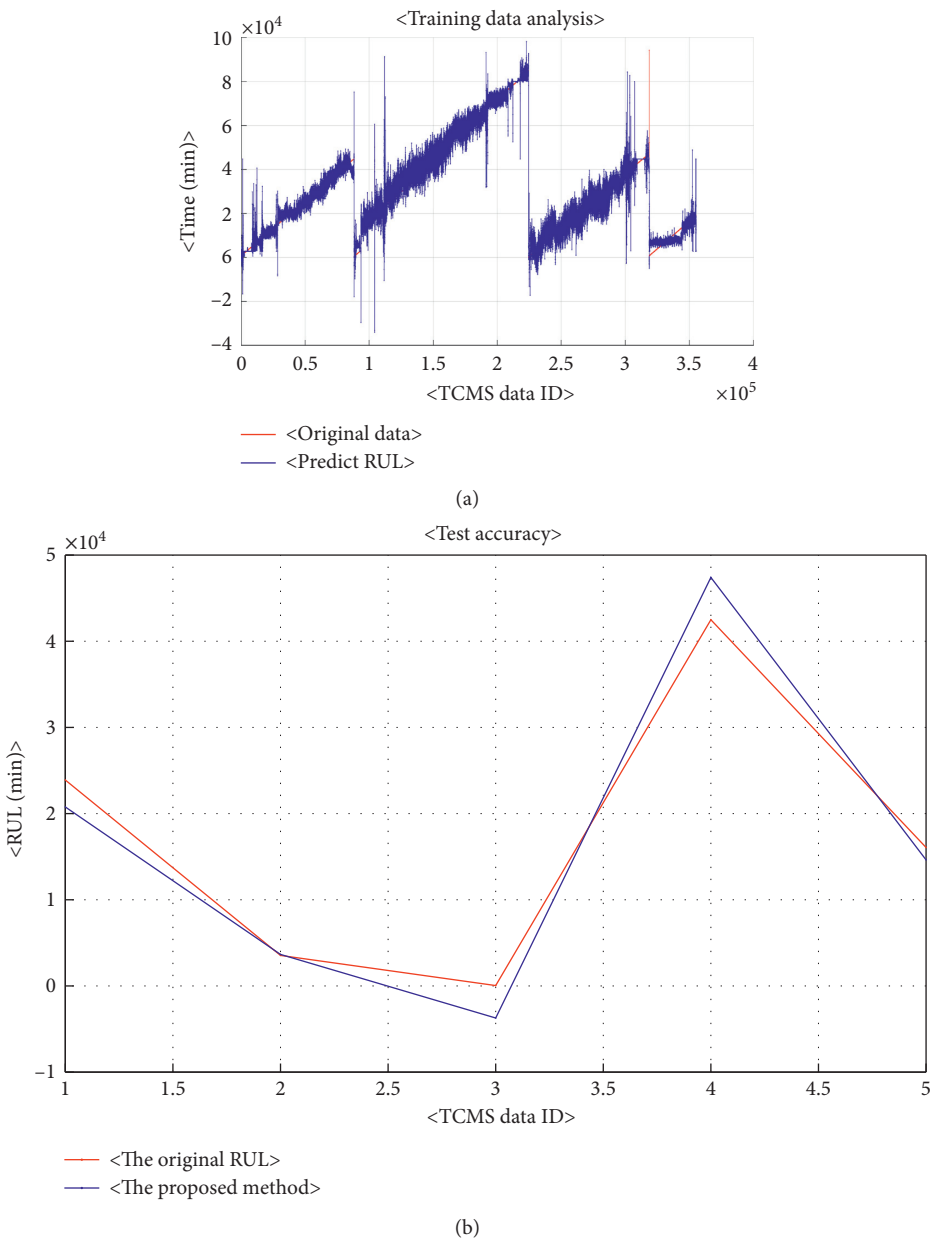


Figure 6: The training and the test accuracies of the proposed RUL prediction (for fault code ID 31). (a) The training accuracy. (b) The test accuracy.

$$\mathrm{RMSE}(\mathrm{RUL}_i) = \sqrt{\frac{\sum_{j=1}^{n}\left(\widehat{\mathrm{RUL}_{i,j}} - \mathrm{RUL}_{i,j}\right)^2}{n}}, \tag{10}$$

$$\mathrm{Accuracy}\ (\%) = \left[1 - \mathrm{RMSE}(\mathrm{RUL}_i)\right] \cdot 100. \tag{11}$$

The test data was sampled from the original TCMS data. As the fault occurrence was very low, the amount of test data was limited. The data-based numerical tests were carried out by comparison with the other methods. Figure 7 shows the test accuracies using the four methods: the proposed method and the other three benchmarking methods shown in Table 7.

As shown in Figure 7, the proposed method had the highest accuracy compared with the other existing methods. Table 9 provides the test accuracy for each method.

The numerical analysis indicates that the missing value issue was critical for the fault prediction using the TCMS data. In addition, the estimation of the missing values strongly influenced the RUL predictions. Using the proposed framework, the RUL prediction system for the 16 train faults was developed as shown in Figure 8. The software program was implemented using MFC© and MATLAB© on Windows 10©. Figure 9 shows the test accuracies of the various TCMS faults.

As shown in Figure 9, the proposed framework and its implemented software resulted in RUL predictions of over 82.07% for all TCMS fault types. Table 10 shows the test

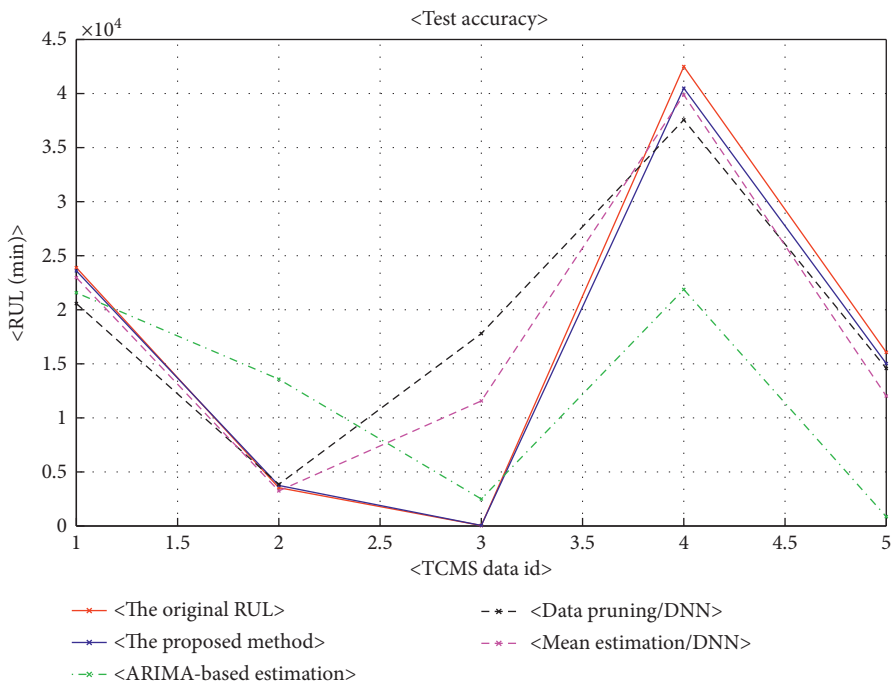


Figure 7: Test accuracies using the four benchmarking methods.

Table 9: Test accuracy for each RUL estimation method (case for fault code ID 31).

RUL prediction method	GAN-based RUL estimation (the proposed method)	ARIMA-based RUL estimation (%)	RUL estimation using "missing value pruning" (%)	RUL estimation using "mean-value estimation" of missing values (%)
Test accuracy	83.5%	56.7	69.7	76.3

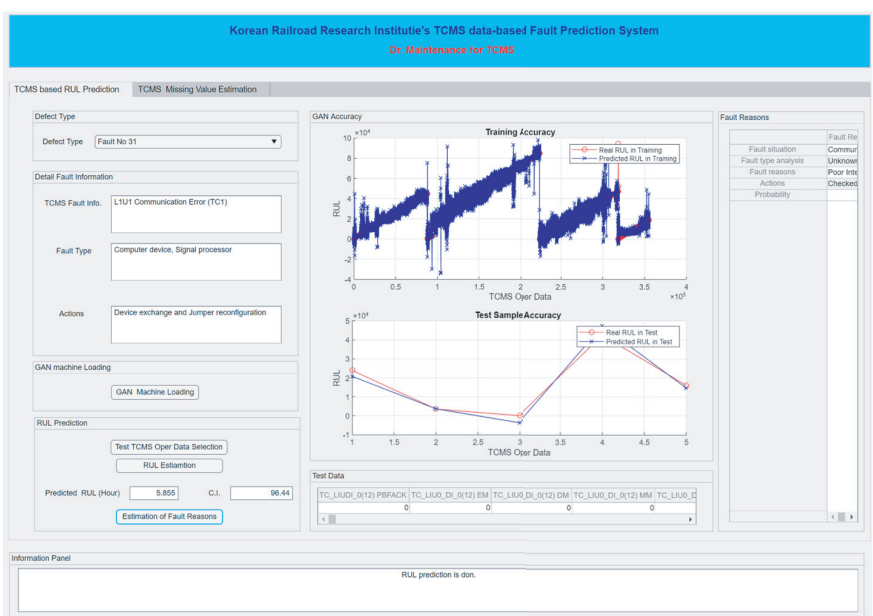


Figure 8: The implementation of GAN-based missing value handling and RUL prediction framework.

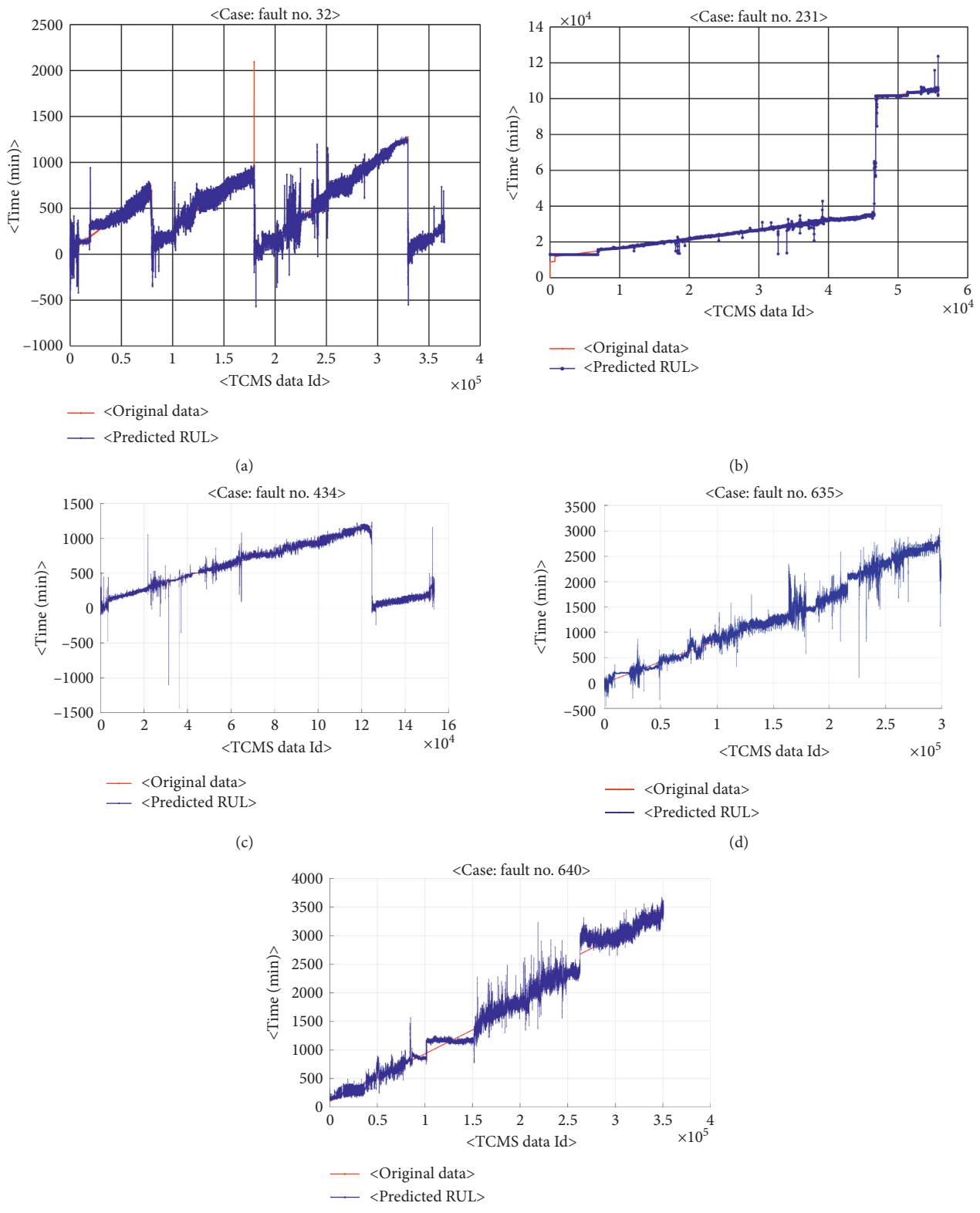
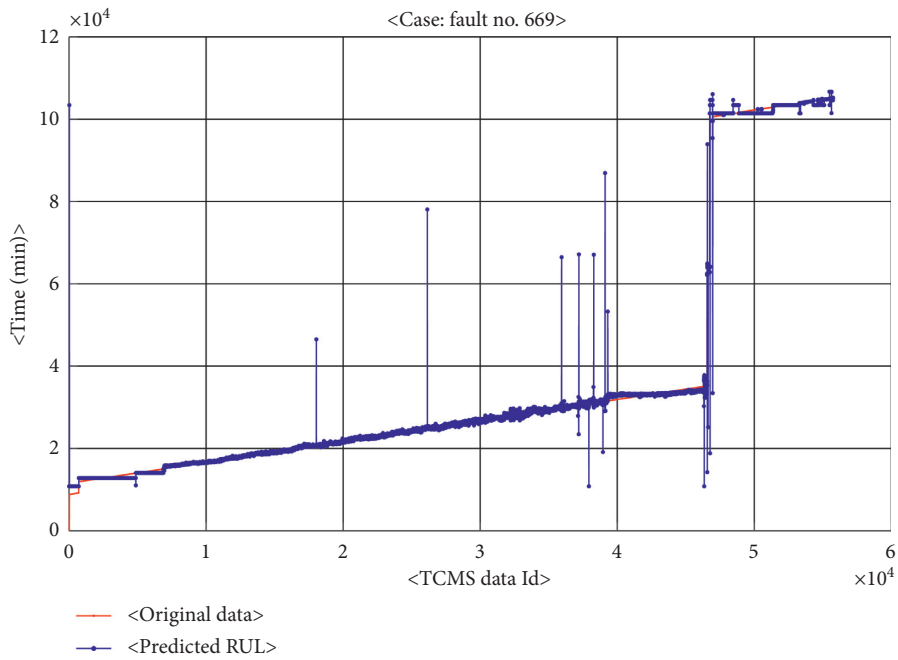


Figure 9: Continued.



(f)

Figure 9: RUL prediction results using the proposed method. (a) Fault code ID 34 (LIU2 communication error in TC0). (b) Fault code ID 231 (SIV inverter malfunction). (c) Fault code ID 434 (break malfunction). (d) Fault code ID 635 (TC MFB malfunction). (e) Fault code ID 640 (main ATC hardware malfunction LIU2). (f) Fault code ID 669 (ATO-ATC communication error).

Table 10: Test accuracy for each defect using the proposed framework.

Fault code id.	Test accuracy	Fault code ID	Test accuracy	Fault code ID	Test accuracy	Fault code ID	Test accuracy (%)
31	83.50	32	88.08	34	88.94	38	82.47
39	86.98	231	99.91	434	96.75	442	95.68
635	85.24	636	85.60	640	92.21	641	89.15
647	93.14	669	97.56	670	82.07	684	80.46

accuracy of the proposed framework for each defect in the TCMS data.

6. Conclusions and Further Study

The transportation maintenance has gained substantial attention owing to its significance in societies. This study focused on the RUL prediction-based railway maintenance framework. While initial railway maintenance concentrated on periodic maintenance framework such as predefined time-based maintenance or distance-based scheduling, railway maintenance has evolved to condition-based maintenance owing to the advancement in monitoring devices and information technologies. This framework detects abnormal status using state-of-the-art sensors in a train system. The monitored signals are transferred to a server, TCMS.

This study proposes a new and effective RUL prediction framework using TCMS data. In general, TCMS data are classified into operation data and alarm data. To predict the remaining life of a certain fault or malfunction, this paper selected 16 faults based on their significance and severity. A deep learning-based mechanism was developed for each fault. Firstly, RUL of the target train fault was extracted using the TCMS alarm data. Then, the data was used as the predicted output of the proposed deep neural network. However, the system has a critical issue, which is common in most sensor-based systems: the existence of missing values. Existence of missing values in TCMS data could be due to various reasons such as sensor malfunction and low life of monitoring modules. Among several estimation methods for replacing missing values, this study used a GAN model to estimate missing values and predict RULs, simultaneously. The developed GAN framework can generate new data that cover missing values using the prediction objectives. Initially, the missing values are estimated using GMM and the estimated data are refined with the proposed GAN framework. In addition, the discriminator in the GAN model has better predictive performances in generating more accurate data. The effectiveness of the proposed maintenance framework was investigated by comparing it with other existing methods. The proposed framework is a new and effective train predictive maintenance framework that addresses missing value issue and predicts fault detection in real time.

For further studies, various optimization and meta-heuristics methods can be applied to the proposed framework. As TCMS data is classified as big data, its learning could take longer time. Moreover, the framework requires comparatively higher computational burden. While the proposed framework uses two deep neural network models for a generator and a discriminator in its GAN module, it is expected that application of several optimization methods could reduce learning time and computation burden. In addition, prediction of each train fault requires a different GAN-based framework. While it has an advantage focusing on the defined fault, real-time prediction may require significant computational burden. To resolve this issue, a new architecture for multiple-fault prediction will be considered in future studies.

Data Availability

The used data is supported by Korea Railroad Research Institute.

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 Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Republic of Korea (grant number: NRF-2018R1D1A3B07047113), and by a research grant from R&D Program of the Korea Railroad Research Institute (KRRI), Republic of Korea.

References

[1] T. Lidén, "Railway infrastructure maintenance - a survey of planning problems and conducted research," *Transportation Research Procedia*, vol. 10, pp. 574–583, 2015.

[2] P. Fraga-Lamas, T. M. Fernandez-Carames, and L. Castedo, "Towards the internet of smart trains: a review on industrial IoT-Connected Railways," *Sensors*, vol. 17, no. 9, pp. 1–44, 2017.

[3] H. Zhao, Z. Huang, and Y. Mei, "High-speed EMU TCMS design and LCC technology research," *Engineering*, vol. 3, no. 1, pp. 122–129, 2017.

[4] J. Han and C. Kim, "A conceptual design of maintenance information system interface for real-time diagnosis of driverless EMU," *Journal of the Korea Academia-Industrial Cooperation Society*, vol. 18, no. 10, pp. 63–68, 2017.

[5] K. Kim, K. Lee, and S. An, "An analysis about consumed energy of electric multiple unit used TCMS data on the condition of safety driving," *Journal of the Korean Society of Safety*, vol. 27, no. 6, pp. 31–42, 2012.

[6] S. Ito, T. Suzuki, K. Suzuki, and K. Suzuki, "Train control and management system technologies for improving safety and maintainability," *Hitachi Review*, vol. 67, no. 7, pp. 52–58, 2018.

[7] K. Sharma, S. Maheshwari, R. Solanki, and V. Khanna, "Railway track breakage detection method using vibration estimation sensor network," in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics*, pp. 2355–2362, New Delhi, India, September 2014.

[8] R. Sireesha, K. B. Ajay, G. Mallikarjunaiah, and K. B. Bharath, "Broken rail detection system using RF technology," *Proceedings of SSRG International Journal of Electronics and Communication Engineering*, vol. 2, no. 4, pp. 11–15, 2015.

[9] H. Lee, "Framework and development of fault detection classification using IoT device and cloud environment," *Journal of Manufacturing Systems*, vol. 43, no. 2, pp. 257–270, 2017.

[10] H. Lee, "Effective dynamic controls strategy of key supplier with multiple downstream manufacturers using industrial Internet of Things and cloud system," *Processes*, vol. 7, no. 3, pp. 1–18, 2019.

[11] Use IoT to advance railway predictive maintenance. (https://www.hitachivantara.com 2020).

[12] F. Corman, S. Kraijema, M. Godjevac, and G. Lodewijks, "Optimizing preventive maintenance policy: a data-driven application for a light rail braking system," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 231, no. 5, pp. 534–545, 2017.

[13] Mckinsey, *Using Analytics to Get European Rail Maintenance on Track*, McKinsey & Company, New York, NY, USA, 2020, https://www.mckinsey.com/.

[14] R. B. Faiz and S. Singh, "Time based predictive maintenance management of UK rail track," in *Proceedings of the 2009 International Conference on Computing, Engineering and Information*, Fullerton, CA, USA, April 2019.

[15] K. Shaikh, I. H. Kalwar, B. S. Chowdhry, K. Kazi, and B. A. Arain, "Modeling and simulation of predictive maintenance scheme for high speed railway vehicles," *Indian Journal of Science and Technology*, vol. 9, no. 1, pp. 1–6, 2016.

[16] C. Letot, P. Dersin, M. Pugnaloni et al., "A data driven degradation-based model for the maintenance of turnouts: a case study," *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 958–963, 2015.

[17] V. Atamuradov, K. Medjaher, P. Dersin, B. Lamoureux, and N. Zerhouni, "Prognostics and health management for maintenance practitioners–review, implementation and tools evaluation," *International Journal of Prognostics and Health Management*, vol. 8, pp. 1–31, 2017.

[18] T. Liden, "Railway infrastructure maintenance–a survey of planning problems and conducted research," *Transportation Research Procedia*, vol. 10, pp. 574–583, 2018.

[19] G. Neil, "On board train control and monitoring systems," in *Proceedings of the 9th Institution of Engineering and Technology Professional Development Course on Electric Traction Systems*, Manchester, UK, November 2006.

[20] Y. Xu, Q. Qiao, R. Wu, and Z. Zhou, "Advanced maintenance cycle optimization of urban transit vehicle," *Advances in Mechanical Engineering*, vol. 11, no. 2, pp. 1–7, 2019.

[21] 2020 Innovative monitoring and predictive maintenance solutions on lightweight wagon, http://newrail.org/innowag/wp-content/uploads/2017/12/INNOWAG_D1.1_Benchmark-market-drivers.pdf.

[22] E. Oh and H. Lee, "An imbalanced data handling framework for industrial big data using Gaussian Process Regression-based Generative Adversarial Network," *Symmetry*, vol. 12, no. 4, pp. 1–19, 2020.

[23] H. Kim and H. Lee, "Fault detect and classification framework for semiconductor manufacturing processes using missing data estimation and Generative Adversary Network," *Journal*

of Korean Institute of Intelligent Systems, vol. 28, no. 4, pp. 393–400, 2018.

[24] H. Kim and H. Lee, "Generative adversarial networks based data generation framework for overcoming imbalanced manufacturing process data," *Journal of Korean Institute of Intelligent Systems*, vol. 29, no. 1, pp. 1–8, 2019.

[25] S. Munirathinam and B. Ramadoss, "Predictive models for equipment fault detection in the semiconductor manufacturing process," *International Journal of Engineering and Technology*, vol. 8, no. 4, pp. 273–285, 2016.

[26] D. Moldovan, T. Cioara, I. Anghel, and I. Salomie, "Machine learning for sensor-based manufacturing process," in *Proceedings of the 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 147–154, Cluj-Napoca, Romania, September 2017.

[27] E. R. Hruschka, E. R. Hruschka, and N. E. F. Ebecken, "Missing values imputation for a clustering genetic algorithm," *Lecture Notes in Computer Science*, vol. 3612, pp. 245–254, 2005.

[28] Y. C. Yuan, "Multiple imputation for missing data: concepts and new development," SAS Institute Inc, Cary, NC, USA, 2019, http://support.sas.com/rnd/app/stat.

[29] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with Applications*, vol. 91, pp. 464–471, 2018.

[30] S. Choo and H. Lee, "Learning framework of multimodal Gaussian-Bernoulli RBM handling real-value input data," *Neurocomputing*, vol. 275, no. 1, pp. 1813–1822, 2018.

[31] G. Casella and R. L. Berger, *Statistical Inference*, Cengage Learning, Boston, MA, USA, 2nd edition, 2002.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8841810, 10 pages
https://doi.org/10.1155/2020/8841810

Research Article

Neural Network-Based Train Identification in Railway Switches and Crossings Using Accelerometer Data

Rostislav Krč,[1] Jan Podroužek,[1] Martina Kratochvílová,[1] Ivan Vukušič,[2,3] and Otto Plášek[2]

[1]*Institute of Computer Aided Engineering and Computer Science, Faculty of Civil Engineering, Brno University of Technology, Brno 602 00, Czech Republic*
[2]*Institute of Railway Structures and Constructions, Faculty of Civil Engineering, Brno University of Technology, Brno 602 00, Czech Republic*
[3]*Výzkumný Ústav Železniční, a.s. (VUZ), Prague 142 00, Czech Republic*

Correspondence should be addressed to Rostislav Krč; rostislav.krc@vutbr.cz

Received 14 September 2020; Revised 23 October 2020; Accepted 12 November 2020; Published 24 November 2020

Academic Editor: Petr Dolezel

This paper aims to analyse possibilities of train type identification in railway switches and crossings (S&C) based on accelerometer data by using contemporary machine learning methods such as neural networks. That is a unique approach since trains have been only identified in a straight track. Accelerometer sensors placed around the S&C structure were the source of input data for subsequent models. Data from four S&C at different locations were considered and various neural network architectures evaluated. The research indicated the feasibility to identify trains in S&C using neural networks from accelerometer data. Models trained at one location are generally transferable to another location despite differences in geometrical parameters, substructure, and direction of passing trains. Other challenges include small dataset and speed variation of the trains that must be considered for accurate identification. Results are obtained using statistical bootstrapping and are presented in a form of confusion matrices.

1. Introduction

Railway switches and crossings (S&C) are important components of railway infrastructure. Dynamic effects of passing trains are higher than in case of a straight track and are affected by factors such as train speed, S&C geometry, fastening stiffness, and substructure material [1]. With increasing traffic and growing demands on the infrastructure, reliability and safety of S&C must be ensured. Large demands on maintenance occur especially on high-speed tracks [2]. Generally, three different maintenance approaches can be applied—corrective, preventive, and predictive [3].

Modern predictive approaches require real-time monitoring and data collection to evaluate S&C condition and apply appropriate countermeasures when needed [4]. Accelerometer or deflection sensors are simple and reliable devices that can be mounted directly in the S&C structure for monitoring the dynamic response. Gradual changes over time for the same train type and speed may indicate an emerging defect in S&C structure [5] and provide an early warning to the infrastructure operators. Therefore, train type must be recognized from the data to evaluate changes in S&C.

Project S-CODE (Switch and Crossing Optimal Design and Evaluation [6]) presented requirements for the next generation of S&C [7] and also introduced *next generation of control, monitoring, and sensing system* that, among others, will be able to determine the type of passing train based on accelerometer data. This system is referred to as Train Identification System (TIS). Recent studies also proposed to utilize machine learning techniques for predictive maintenance [8]. Train type was already successfully identified in a straight track [9]. Identification of trains in S&C is a more challenging task as more factors affecting sensor measurements must be considered.

Data can be obtained either from sensors mounted on trains or track. For successful train identification, it is important to recognize defects on trains such as flat wheels and not consider data from defected trains in S&C evaluation. Train defects such as wheel flats have been already detected by sensors mounted on trains [10] or track [11]. Critical samples containing defected wheels can be identified from the accelerometer signal by state-of-the-art pattern recognition techniques [12].

Machine learning methods can be used with benefit for processing a large amount of data. Methods such as support vector machines (SVMs) have already been incorporated for condition monitoring of railway infrastructure [13]. In this paper, neural networks are used for train identification as they are suitable for time series classification problems [14, 15]. Once trained, neural networks are also advantageous in terms of performance which may be useful for future in situ TIS.

The aim of this paper is to introduce possibilities of train type identification directly in S&C using neural networks and accelerometer data. This approach is unique and has not been attempted to date. Two locations and four S&C are considered, and several use case scenarios are presented in order to evaluate the transferability of machine learning models between different locations. Results for multiple train classes as well as various neural network architectures are discussed.

2. Data and Methods

2.1. Data Acquisition. Data used for train identification were obtained by in situ measurements from multiple accelerometer sensors placed in different positions around the common crossing of the S&C. The common crossing contained no movable parts. Therefore, passing trains caused increased acceleration impulses due to interruption of the rail continuity as wheels of the train hit the crossing nose. In a case of a movable crossing that is used in some S&C designs especially for high-speed tracks, these impulses would be lower but still detectable [16].

The full dataset contains signals from 6 single-axis accelerometers in Z-direction, 2 three-axis accelerometers in X, Y, and Z directions, and 8 displacement sensors in Z-direction as shown in Figure 1. The sampling frequency of the sensors was 10 kHz. Sensors were placed either on ballast bed, sleeper, or directly on a rail near the crossing nose.

2.2. Characteristics of Locomotive Classes. Seven locomotive classes were chosen for identification as they vary in geometry, weight, or undercarriage stiffness: class 150/151 (denoted as 151), classes 162/163 (denoted as 163), class 362/363 (denoted as 363), class 380, Pendolino 680 (denoted as 680), Stadler 480 (denoted as 480), and class Siemens ES64U2/ES64U4 (denoted as Taurus). Geometrical parameters and weights for each class are shown in Table 1.

Data were obtained from two nearby locations on the same railway corridor in the Czech Republic: Choceň (referred to as Location 1) and Ústí nad Orlicí (referred to as Location 2). Two S&C were present in each location and their parameters differed between locations. Both S&C in Location 1 had different geometry (suitable for higher speeds), substructure parameters, and also an opposite direction of train passages compared to the S&C in Location 2. Another difference was that trains with locomotive class 363 had lower mean speed in Location 2 as they stopped in a nearby station. The speed of the trains was measured by a radar velocity gun with ±2 km/h accuracy. Measurements for each locomotive class and their speeds are listed in Table 2 for Location 1 and Table 3 for Location 2.

2.3. Localization of the Locomotive Part. Locomotive part of the accelerometer signal was used for the identification since locomotives are usually better maintained compared to the regular carriages. The variance of locomotive weights is also lower. Approaches used for locomotive localization from the whole signal were based on peak detection. Root mean square (RMS) value was calculated by equation (1) using a sliding window of size $d = n_1 - n_2$ for peak localization. Grouping of nearby peaks was done by mean shift clustering with bandwidth parameter α:

$$\mathrm{RMS}_{n_2,n_1} = \sqrt{\frac{1}{n_2 - n_1} \sum_{n_1}^{n_2} |x(n)|^2}. \quad (1)$$

The size of the sliding window for RMS was chosen to $d = 0.02$ s. Peaks were then limited by a minimal amplitude value that was calculated dynamically using quantile of the whole signal between $q_{\mathrm{lim}} = 0.85 \sim 0.95$. Mean shift clustering with bandwidth parameter $\alpha = 0.03 \sim 0.033$ s distance was applied in order to group nearby peaks. All parameters were chosen empirically based on mean train speed. These methods served only for preprocessing of the given dataset and are not the aim of this research.

Each peak in an accelerometer signal represents an axle of a train and a two-peak group represents a bogie. Therefore, the signal can be divided into four-peak groups where the first group represents a locomotive which is followed by carriages as subsequent groups. This algorithm proved itself useful in data preprocessing and automatic extraction of the locomotive part of the signal as it was applied on a dataset which contained mostly signals with low levels of noise.

Example of an accelerometer signal generated by train with a locomotive of class 380 at speed 162 km/h passing through a S&C is shown in Figure 2. All axles of the train can be easily recognized as peaks in the signal. Detail of the locomotive part of this signal is shown in Figure 3.

2.4. Methodology for Classification. The high cost of corrective maintenance and risk of accidents require a robust solution for train type identification as it will be part of the S&C real-time monitoring system. The S-CODE project was proposed to incorporate accelerometer signals to determine the type of passing train [6]. Accelerometer sensors will be mounted in situ in the S&C structure and it is expected that a

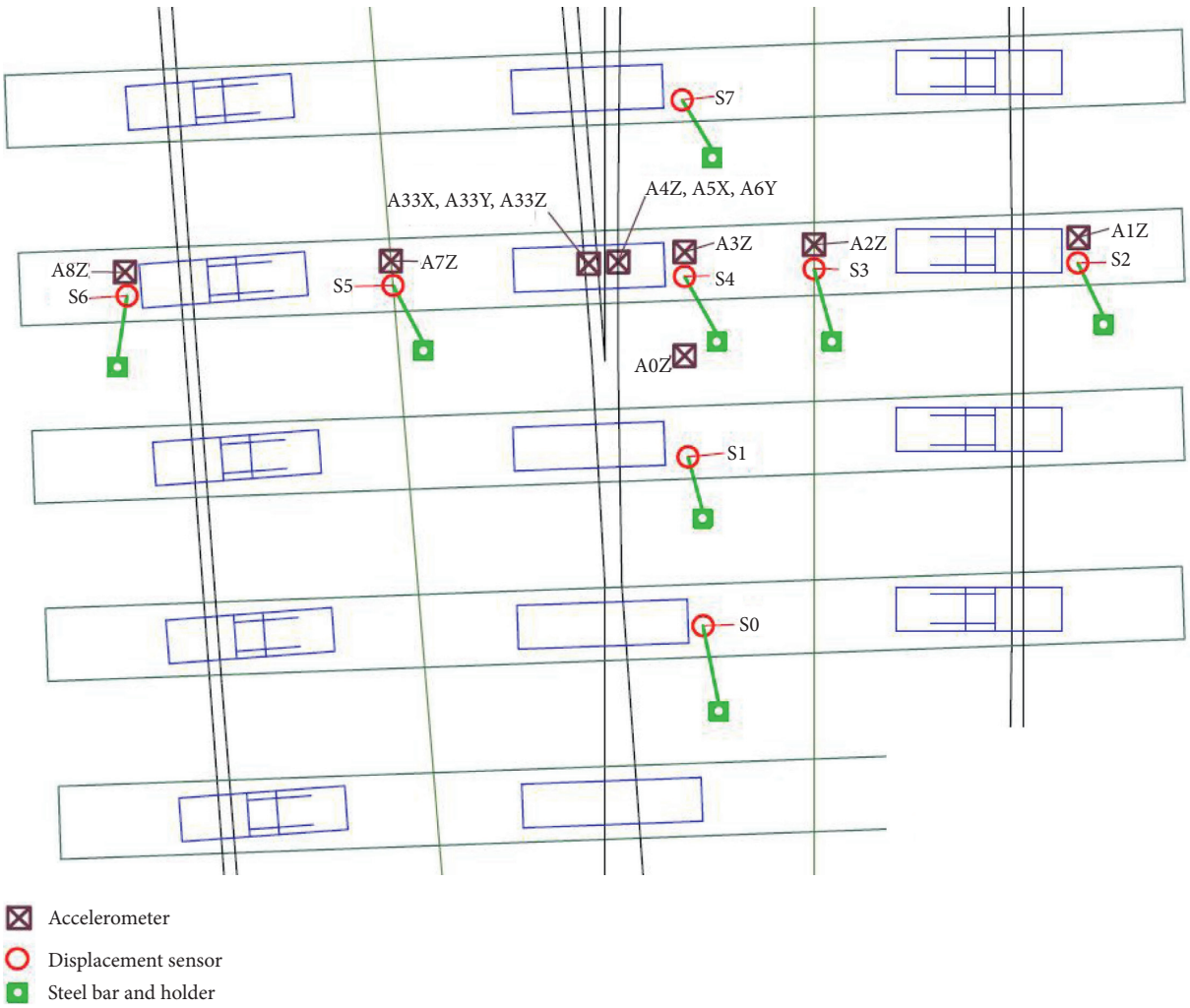


Figure 1: Sensor placement layout around the common crossing of the S&C used for data acquisition.

Table 1: Geometrical parameters and weights of different locomotive classes.

Locomotive class	151	163	363	380	480	680	Taurus
Distance between pivots (m)	8.3	8.3	8.3	8.7	16.0	19.0	9.9
Axle spacing (m)	3.2	3.2	3.2	2.5	2.7	2.7	3.0
Weight (t)	82.0	84.0	87.0	86.0	150.0[1]	57.0	87.0

[1]Total weight of the whole five-car train.

Table 2: Location 1: measured locomotives and their speeds.

Locomotive class	151	163	363	380	480	680	Taurus
Number of measurements (-)	10	8	8	9	6	7	6
Mean speed (km/h)	133.2	106.5	129.6	147.4	159.3	154.4	145.3
Speed standard deviation (km/h)	15.5	35.5	13.0	13.0	4.4	4.7	9.8

large amount of data will be collected over time, so appropriate methods must be chosen for further data processing.

As stated in [13], machine learning methods, such as support vector machines (SVMs), are often used for monitoring and evaluation of the condition of railway infrastructure components [17] or for train defect detection from sensor data [11]. Using neural network-based models for time series classification is a common problem [18], and recent research mostly focuses on developing novel network

Table 3: Location 2: measured locomotives and their speeds.

Locomotive class	151	363	380	480	680
Number of measurements (-)	10	8	12	12	12
Mean speed (km/h)	122.0	91.9	128.1	147.0	128.5
Speed standard deviation (km/h)	5.7	14.8	4.9	12.7	4.3

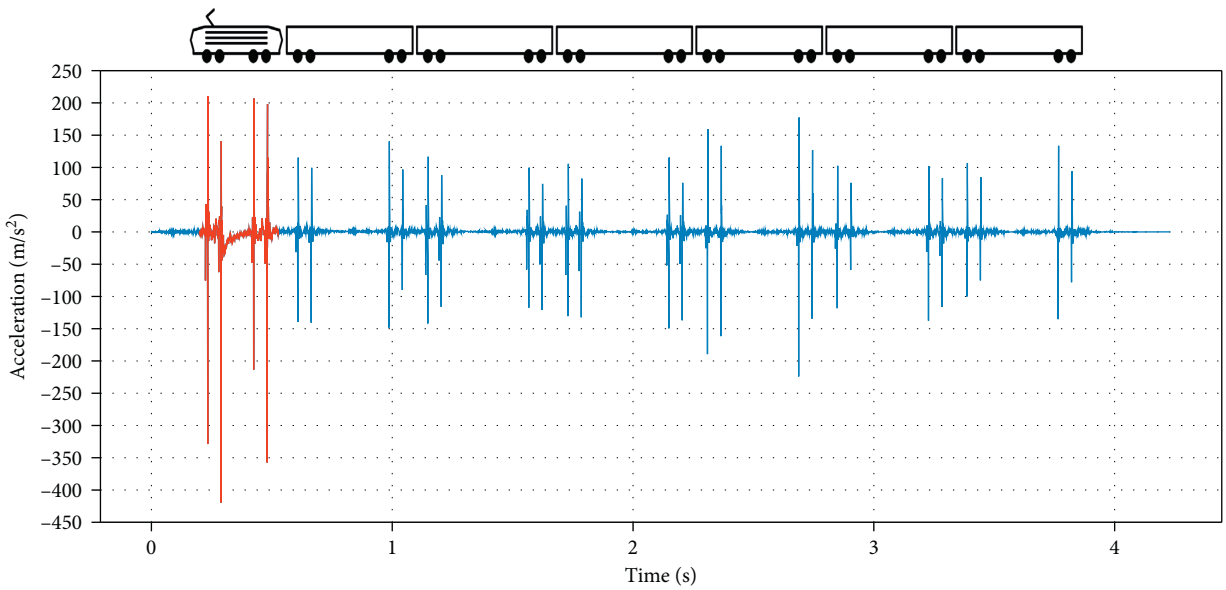


Figure 2: Accelerometer signal (Z-axis) during train passage over a crossing. The locomotive part is highlighted in red and shown in detail in Figure 3.

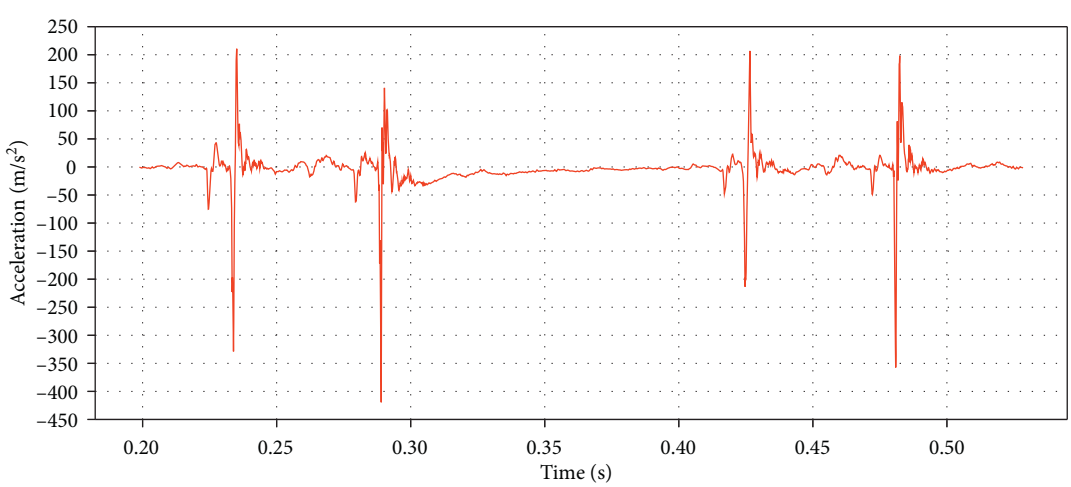


Figure 3: Detail of the locomotive part of the signal shown in Figure 2.

architectures such as modifications of residual neural network (ResNet) [19]. Convolutional neural networks are often used for the classification of time series data with an outstanding performance [20, 21] and are also widely used for the classification of accelerometer data and human activity recognition [14, 22]. In railway engineering, deep neural networks were successfully applied in areas such as fault diagnosis on trains [23] or for rail degradation prediction [24].

Given the high complexity of the train type identification problem in S&C, multiple neural network architectures will be examined in this paper in order to find an optimal design.

2.5. Neural Network Design and Training. Six different neural network architectures were evaluated for locomotive classification. Four multilayer perceptrons with either one

(MLP1), two (MLP2), or three (MLP3a, MLP3b) fully connected hidden layers. Hidden layer size was set to 100 neurons in all cases except MLP3b where 500 neurons were used. All perceptron models employed rectified linear activation function (ReLU) between layers except the output layer where softmax activation was applied. Using the softmax activation function in the output layer is a common practice [25] which has an advantage that the vector of output probabilities sums to 1.

A convolutional neural network (CNN) consisted of a convolutional layer with 64 filters of length 5 followed by a max-pooling layer of length 5 and a hidden layer of size 100. ReLU was used as an activation function between layers and softmax activation at the output.

The sixth and final architecture was a long short-term memory recurrent neural network (LSTM) with one LSTM layer with 50 hidden states followed by a fully connected layer with output softmax activation function.

The input size for all models was set to 1000, and the output size was the number of classified train types (i.e., 5 and 7). The training was done in 12 epochs and data were forwarded through the model in batches of size 4. The learning rate of the models was fixed to 0.001. Adam optimizer was selected for automatic differentiation [26], and mean squared error was used as a loss function. The number of trainable parameters and the number of layers for different neural network architectures are presented in Table 4.

2.6. Normalization of Input Accelerometer Signals. Specific features from the data can be selected as input to the neural networks to decrease complexity and improve training times. However, a whole accelerometer signal may be used without a need for extensive and domain-specific preprocessing. This approach also removes bias due to manually selected features [18] and improves performance, especially for in situ device.

In the first step, signals were normalized in both X- and Y-axes to prevent locomotive misclassification for different train speeds. The number of samples in available locomotive signals spanned between $1 \cdot 10^3$ and $1 \cdot 10^4$ depending on the sampling frequency of the sensors, train speed, and locomotive geometry. In the X-axis, signals were resampled to the input size which was chosen to 1000. This number of samples is sufficient as it preserves enough information with a lower number of samples than in the original signal (see Figure 4). In the Y-axis signals were normalized between values −1 and 1.

2.7. Use Case Scenarios. Four accelerometer channels A0Z, A2Z, A3Z, and A7Z were selected for train identification as they were similar in terms of phase shift and noise. Sensors A2Z, A3Z, and A7Z were placed on a sleeper under the crossing nose and sensor A0Z was placed in a ballast bed nearby as shown in Figure 1. These four channels were used separately in order to augment data and increase its variability as the sensors can generally be placed in arbitrary position around the crossing nose. The full dataset contained 108 train measurements from Location 1 and Location 2 giving in a total of 432 samples. To evaluate classification models for a different variety of data, these two locations were considered both independently and together, using only locomotive classes present in both locations (5 classes).

Four use case scenarios were considered as shown in Table 5. In scenarios A and B, the models were trained on all the samples from Location 1 and Location 2, respectively. In scenario C, the data from these two locations were combined. Size of the dataset remained relatively small despite using four accelerometer channels independently. Therefore, the bootstrapping technique [27] was utilized for scenarios A, B, and C in order to produce statistically relevant results. 10 models were trained and tested for each neural network architecture and each scenario, and the results were averaged to evaluate the overall performance of the given architecture [28]. For every model, the scenario dataset was shuffled and split in the way that at least two locomotive passages (i.e., 8 samples) for each class were available for testing.

Finally, the use case scenario D used data from Location 2 for training and the data from Location 1 for testing. This scenario aimed to evaluate a situation when the model for train identification is trained on the currently available data and then applied to another S&C.

3. Results

Substantial differences of classification accuracy between the use case scenarios, locomotive classes, and neural network architectures were observed due to factors such as the variance of train speeds, undercarriage geometry, or dynamic response of S&C structure. Despite these factors, the accuracy of the presented models is still relatively high compared to random classification.

Baseline accuracy (random classification) for scenario A is 14.3% and for scenarios B to D is 20.0% and is given by the inverse of the number of classified classes. Mean model accuracy for different scenarios spanned between 52.3% and 80.6% and is presented in Table 6 and Figure 5. The difference in the mean accuracy in the considered two locations (scenarios A and B) was 28.3% and has to be addressed to the higher data variability in Location 1 as more locomotive classes were classified and also the train speeds were more variable. Training models on data from one location and testing on the other (scenario D) resulted in a mean accuracy of 55.0%. Combining data from both locations together (scenario C) exhibited a mean accuracy of 72.9%. Confusion matrices were used for the evaluation of results.

Differences can also be observed between different neural network architectures (Table 6 and Figure 5). The flexibility of models varies as the number of trainable parameters differs (see Table 4). CNN shows the best accuracy in all scenarios compared to the other models since the convolutional layer enhances the ability of feature recognition in time series data. This architecture also contains the largest number of trainable parameters. All multilayer perceptrons (models MLP1, MLP2, MLP3a, and MLP3b) have only low variance in accuracy and with slightly decreasing trend for deeper architectures. Relatively poor mean accuracy was observed in LSTM due to difficulties in

Table 4: Number of layers and number of trainable parameters for the evaluated neural network architectures.

Model	MLP1	MLP2	MLP3a	MLP3b	CNN	LSTM
Number of layers	2	3	4	4	4	2
Number of trainable parameters	100 605	110 705	120 805	1 000 005	1 280 989	260 605

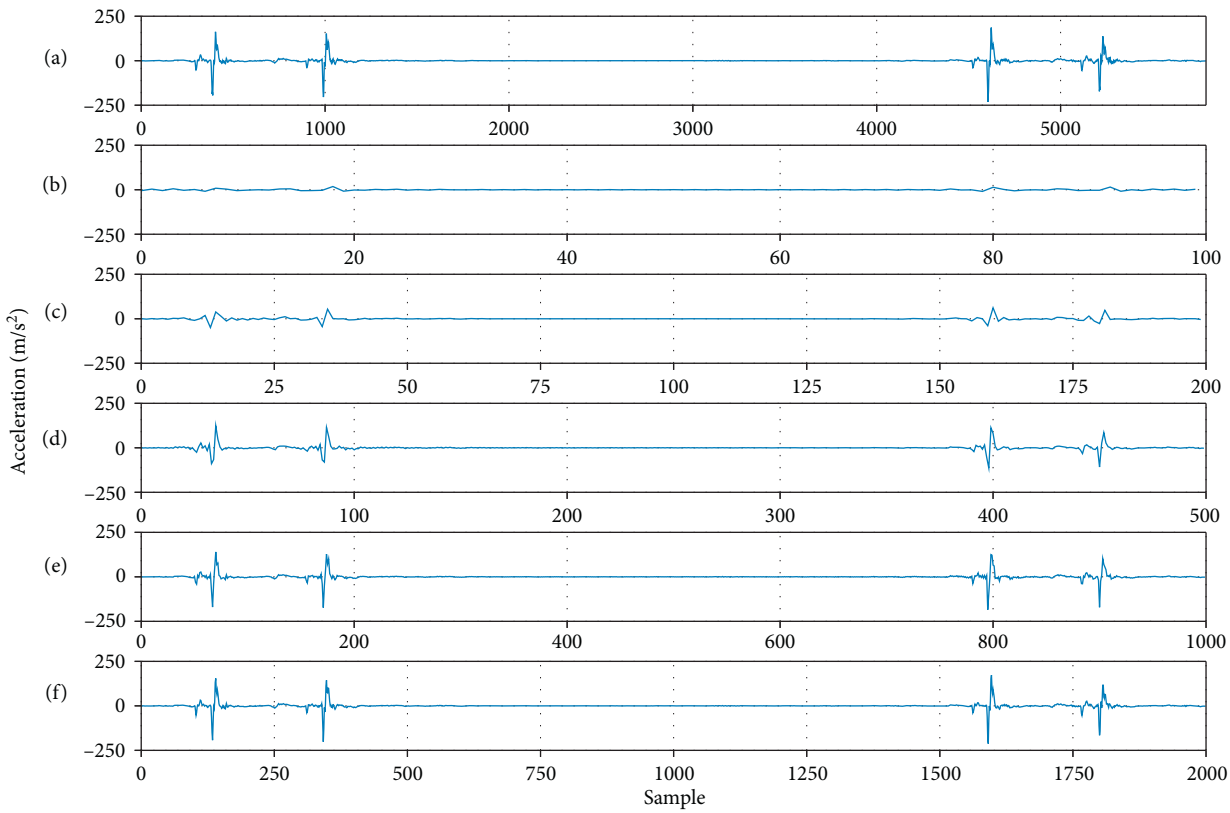


Figure 4: Effect of signal resampling to a different number of samples. (a) Original signal with 5791 samples. (b) Resampled to 100 samples. (c) Resampled to 200 samples. (d) Resampled to 500 samples. (e) Resampled to 1000 samples. (f) Resampled to 2000 samples.

Table 5: Setup of evaluated use case scenarios.

Scenario	Location	Bootstrapping (no. of repeats)	No. of classes	Dataset size	Training size	Testing size
A	Location 1	Yes (30)	7	216	152	64
B	Location 2	Yes (30)	5	216	164	52
C	Location 1 and 2—mixed	Yes (30)	5	376	308	68
D	Location 2 (training), Location 1 (testing)	No	5	376	216	160

Table 6: Mean accuracy (with standard deviation if applicable) for different neural network models and different scenarios. Baseline (random) accuracy is denoted as "Base." Visualization of this table is shown in Figure 5.

Model/Scenario	Base (%)	Mean (%)	MLP1 (%)	MLP2 (%)	MLP3a (%)	MLP3b (%)	CNN (%)	LSTM (%)
A	14.3	52.3 ± 7.9	50.9 ± 4.4	52.3 ± 8.2	51.4 ± 7.0	49.5 ± 8.2	**60.0 ± 6.3**	49.7 ± 7.2
B	20.0	80.6 ± 12.0	82.9 ± 5.7	87.3 ± 7.0	83.3 ± 6.0	81.2 ± 6.0	**89.2 ± 6.9**	59.8 ± 9.7
C	20.0	72.9 ± 9.9	76.2 ± 7.1	74.1 ± 5.2	73.7 ± 9.2	73.5 ± 4.9	**80.6 ± 6.9**	59.3 ± 9.8
D	20.0	55.0	57.5	58.8	53.7	53.1	**72.5**	34.4

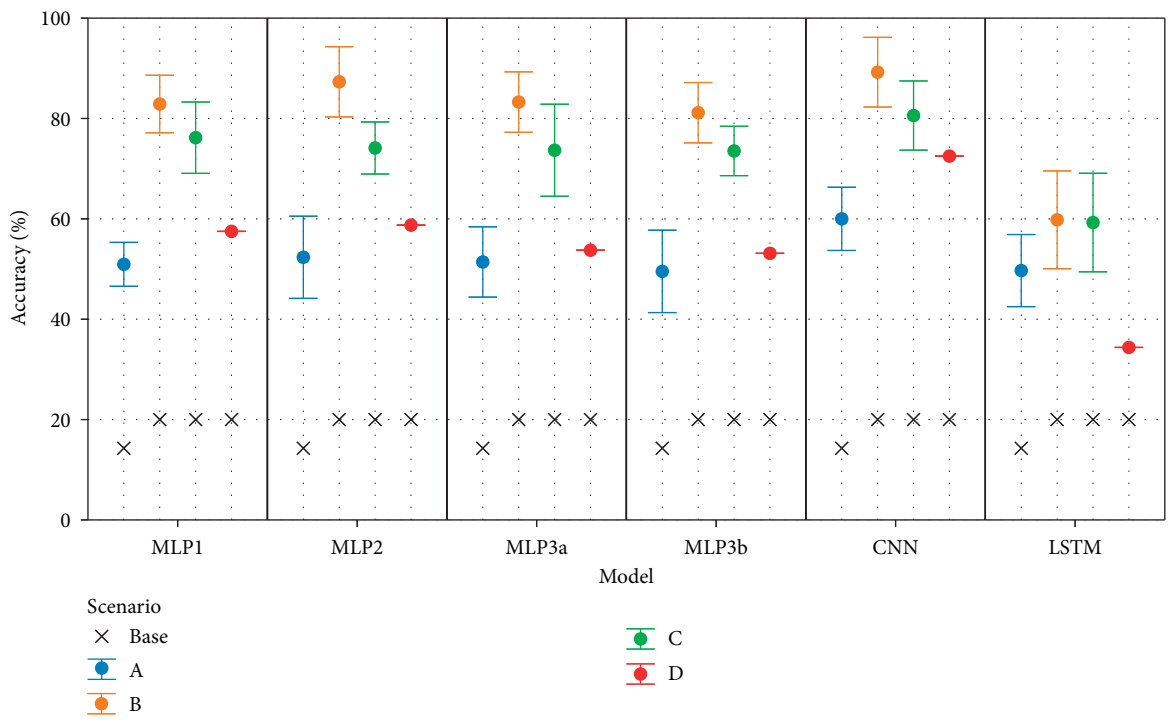


Figure 5: Mean model accuracies with standard deviation (if applicable) for different use case scenarios. Baseline (random) accuracy is denoted as "Base" and is marked in black.

the training process. Mean confusion matrices for the most accurate CNN architecture are presented in Figures 6–9.

Locomotive classes were also classified with varying accuracy. Pendolino 680, Stadler 480, and class 380 were identified with the highest mean accuracy due to their specific undercarriage geometry. On the contrary, mean accuracies in scenario A for the three mutually geometrically similar classes 151, 163, and 363 were lower. Differences in the classification accuracy for the same locomotive classes in different scenarios are to be addressed to the variance of speed. An overview of the mean accuracy of classification for each locomotive class is shown in Table 7.

4. Discussion

Results showed differences in accuracy for different scenarios, locomotives, and machine learning models which can be addressed to factors such as complex dynamic interaction of the train and S&C structure, multiple locomotive classes, similarities in locomotive undercarriage geometries, speed variance, and a relatively small amount of training data. The test scenario C that used data from one location for training and the other location for testing presented that neural network-based classifiers are generally transferable to S&C in different locations. Nevertheless, the model performance has to be improved by using a larger training dataset and more advanced architectures of the neural networks. Additionally, high uncertainty in case of trains with high-speed variance requires partitioning trains with different speeds into separate classes.

The highest classification accuracy of CNN was expected since it is the most commonly used architecture for this type of problem [18]. On the other hand, the lowest accuracy of LSTM compared to the other evaluated models may be attributed to the long input sequence as this architecture is generally suitable for time series classification [29]. Adding a convolutional layer to LSTM may also increase its accuracy as this architecture was successfully applied in a number of time series classification or prediction problems [30, 31].

Trains with different undercarriage geometry were identified with the highest accuracy contrary to the trains with similar geometry that were often mutually misclassified. Large speed variability should also be addressed for the poor classification accuracy for class 363.

It is expected that more accelerometer data from train passages through S&C will be available in the future. Advanced network architectures such as LSTM with convolutional layers [30] or ResNet [19] will be examined as well as more refined optimization of hyperparameters. Also, data augmentation techniques can be employed to increase dataset size and variability [32]. Another possible solution is to use transfer learning [22] and utilize a large amount of data available in other industries. Here, machine learning models can be trained on similar time series data and then fine-tuned for the locomotive classification problem.

The ultimate goal is to develop a full-featured solution for locomotive identification in order to evaluate changes in the dynamic response of S&C for the same train types and speeds as well as to detect defected trains and exclude them from the dataset to improve classification accuracy.

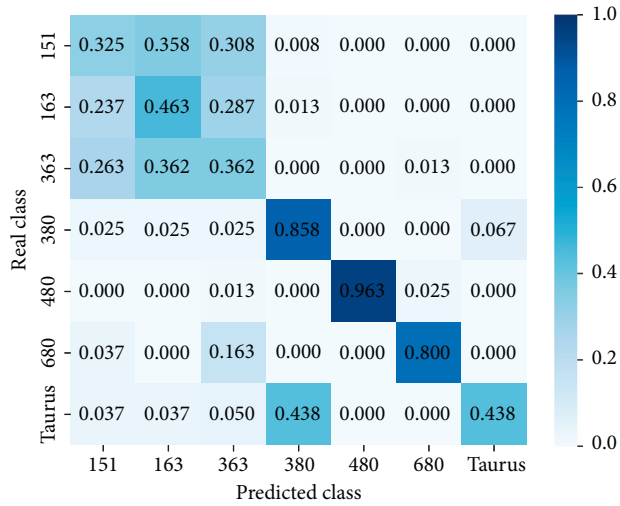


Figure 6: Mean confusion matrix for the best performing CNN for scenario A.

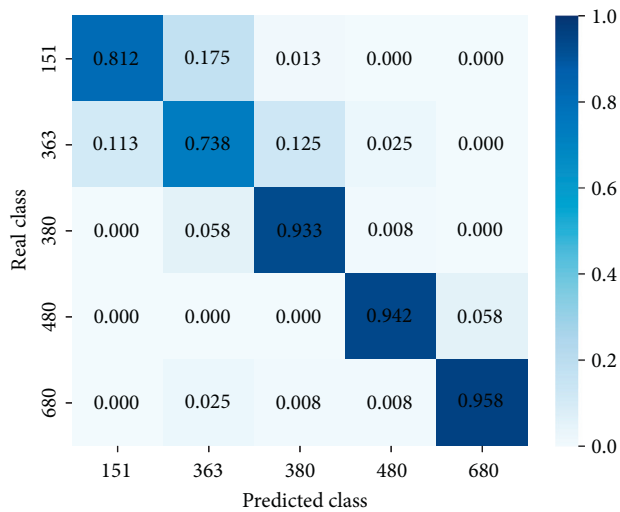


Figure 7: Mean confusion matrix for the best performing CNN for scenario B.

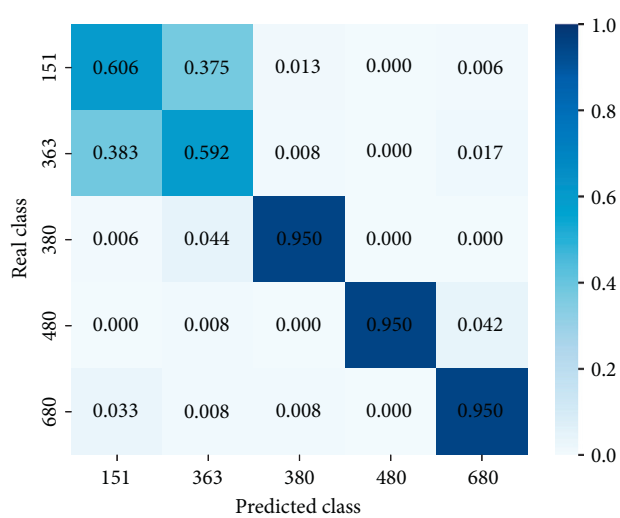


Figure 8: Mean confusion matrix for the best performing CNN for scenario C.

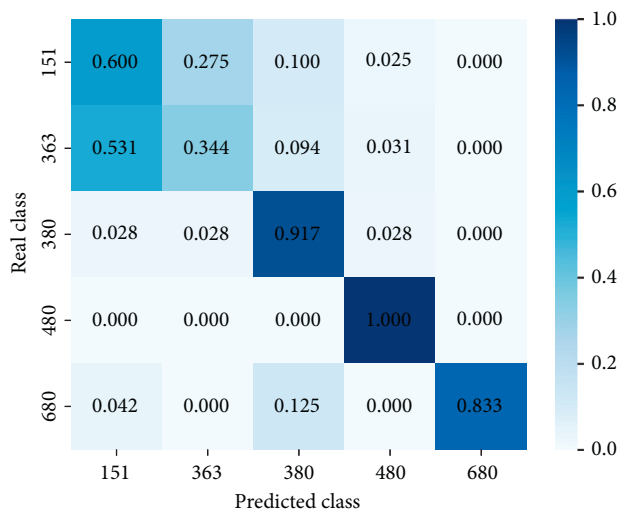


Figure 9: Confusion matrix for the best performing CNN for scenario D.

Table 7: Mean accuracy for different locomotive classes.

Locomotive class/scenario	151 (%)	163 (%)	363 (%)	380 (%)	480 (%)	680 (%)	Taurus (%)
A	30.1	37.5	38.8	61.0	87.9	73.8	44.0
B	81.5	—	47.7	88.7	82.2	92.2	—
C	59.3	—	46.3	87.7	81.3	89.6	—
D	50.4	—	17.2	58.8	81.0	77.1	—

5. Conclusions

Train identification based on accelerometer data in S&C using different neural network architectures was presented in this paper. The most important findings can be summarized as follows:

(i) Train type identification in S&C is feasible despite the increased complexity of the problem compared to a straight track.

(ii) Transferability of machine learning models between different locations is also possible. Models can be trained on data from one location and then applied to another, previously unseen location, with relatively high classification accuracy in spite of differences in S&C parameters. However, both locations evaluated in this paper are positioned on one railway corridor. It is therefore desirable to further verify the transferability of models between unrelated locations.

(iii) Accelerometer signals can be classified without a need for manual feature selection with respect to the limited computational capacity of the in situ device.

To enhance the robustness of evaluated models, only the locomotive part of the signal was used as locomotives are less variable in terms of weight and wheel geometry. However, locomotives with largely different speeds were incorrectly classified despite normalization of input data. Grouping of locomotives into speed categories is required in order to improve classification accuracy. Additionally, defected trains must be identified in advance and excluded from the dataset for successful train identification and subsequent evaluation of the dynamic response of S&C.

Comparison of four use case scenarios and six neural network architectures showed higher model performance for data with lower variability and vice versa. The best performing convolutional neural network proved to be a suitable baseline architecture for the locomotive classification problem. In further research, more advanced neural network architectures, as well as hyperparameter optimization, will be investigated.

Data Availability

The data used in this work were provided exclusively by Správa železnic, the national railway infrastructure manager of the Czech Republic. Data can be provided on demand at the e-mail address info@spravazeleznic.cz.

Conflicts of Interest

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

Acknowledgments

Financial support provided by the Technology Agency of the Czech Republic under the projects *Turnout 4.0* (CK01000091) and *Efficient spacetime predictions using machine learning methods* (TJ04000232) as well as the

support of the project *Smart sensoric system for railways* (FAST/FSI-J-20-6265) is greatly acknowledged.

References

[1] I. Vukušič, D. Vukušičová, K. Zaplatílek, J. Podroužek, J. Apeltauer, and M. Kratochvílová, "Dynamic effects diagnosis in railway switches and crossings within the S-code project," *Scientific and Technical Proceedings of Správa Železnic*, vol. 1, pp. 1–26, 2019.

[2] C. Zhang, Y. Gao, W. Li, L. Yang, and Z. Gao, "Robust train scheduling problem with optimized maintenance planning on high-speed railway corridors: the China case," *Journal of Advanced Transportation*, vol. 2018, Article ID 6157192, 16 pages, 2018.

[3] B. Dhillon, *Engineering Maintenance: A Modern Approach*, CRC Press, Boca Raton, Florida, USA, 2002.

[4] S. Huang, F. Zhang, R. Yu, W. Chen, F. Hu, and D. Dong, "Turnout fault diagnosis through dynamic time warping and signal normalization," *Journal of Advanced Transportation*, vol. 2017, Article ID 3192967, 8 pages, 2017.

[5] M. Sysyn, U. Gerber, O. Nabochenko, Y. Li, and V. Kovalchuk, "Indicators for common crossing structural health monitoring with track-side inertial measurements," *Acta Polytechnica (Prague, Czech Republic : 1992)*, vol. 59, no. 2, pp. 170–181, 2019.

[6] S-Code, http://www.s-code.info/about.

[7] O. Plasek, L. Raif, I. Vukusic, V. Salajka, and J. Zelenka, "Design of new generation of switches and crossings," in *Proceedings of the Conference on Future Trends in Civil Engineering 2019*, vol. 1, pp. 277–301, 2019.

[8] Z. Allah Bukhsh, A. Saeed, I. Stipanovic, and A. G. Doree, "Predictive maintenance using tree-based classification techniques: a case of railway switches," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 35–54, 2019.

[9] E. Berlin and K. Van Laerhoven, "Sensor networks for railway monitoring: detecting trains from their distributed vibration footprints," in *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Cambridge, MA, USA, May 2013.

[10] B. Liang, S. D. Iwnicki, Y. Zhao, and D. Crosbee, "Railway wheel-flat and rail surface defect modelling and analysis by time-frequency techniques," *Vehicle System Dynamics*, vol. 51, no. 9, pp. 1403–1421, 2013.

[11] G. Krummenacher, C. S. Ong, S. Koller, S. Kobayashi, and J. M. Buhmann, "Wheel defect detection with machine learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1176–1187, 2018.

[12] J. Podrouzek, C. Bucher, and G. Deodatis, "Identification of critical samples of stochastic processes towards feasible structural reliability applications," *Structural Safety*, vol. 47, pp. 39–47, 2014.

[13] M. Hamadache, S. Dutta, O. Olaby, R. Ambur, E. Stewart, and R. Dixon, "On the fault detection and diagnosis of railway switch and crossing systems: an overview," *Applied Sciences*, vol. 9, no. 23, p. 5129, 2019.

[14] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Applied Soft Computing*, vol. 62, pp. 915–922, 2018.

[15] S. Vernekar, S. Nair, D. Vijaysenan, and R. Ranjan, "A novel approach for classification of normal/abnormal phonocardiogram recordings using temporal signal analysis and machine learning," in *Proceedings of the 2016 Computing in Cardiology Conference*, Vancouver, Canada, September 2016.

[16] M. Z. Hamarat, S. Kaewunruen, and M. Papaelias, "Contact conditions over turnout crossing noses," *IOP Conference Series Materials Science and Engineering*, vol. 471, no. 6, pp. 1–12, 2019.

[17] H. Tsunashima, "Condition monitoring of railway tracks from car-body vibration using a machine learning technique," *Applied Sciences*, vol. 9, no. 13, 2019.

[18] H. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[19] J. Wu, Z. Zhang, Y. Ji, S. Li, and L. Lin, "A ResNet with GA-based structure optimization for robust time series classification," in *Proceedings of the 2019 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering*, Hangzhou, China, April 2019.

[20] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4788–4797, 2019.

[21] B. Qian, Y. Xiao, Z. Zheng et al., "Dynamic multi-scale convolutional neural network for time series classification," *IEEE Access*, vol. 8, p. 1, 2020.

[22] Recent Research From Swiss Federal Institute Of Technology Highlight Findings In Convolutional Neural Networks (Real-Time Human Activity Recognition From Accelerometer Data Using Convolutional Neural Networks), (Report), Journal of Engineering, 2018.

[23] H. Hu, B. Tang, X. Gong, W. Wei, and H. Wang, "Intelligent fault diagnosis of the high-speed train with big data based on deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2106–2116, 2017.

[24] A. Falamarzi, S. Moridpour, M. Nazem, and R. Hesami, "Rail degradation prediction models for tram system: Melbourne case study," *Journal of Advanced Transportation*, vol. 2018, Article ID 6340504, 8 pages, 2018.

[25] I. Goodfellow, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[26] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2017, https://arxiv.org/abs/1412.6980.

[27] R. W. Johnson, "An introduction to the bootstrap," *Teaching Statistics*, vol. 23, no. 2, pp. 49–54, 2001.

[28] Y. Xu and R. Goodacre, "On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning," *Journal of Analysis and Testing*, vol. 2, no. 3, pp. 249–262, 2018.

[29] Z. Lipton, D. Kale, and R. Wetzel, "Phenotyping of clinical time series with LSTM recurrent neural networks," 2017, https://arxiv.org/abs/1510.07641.

[30] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[31] C.-J. Huang, "A deep CNN-LSTM model for particulate matter (PM2.5) forecasting in Smart cities," *Sensors*, vol. 18, no. 7, p. 2220, 2018.

[32] G. Forestier, J. Weber, and P.-A. Muller, "Data augmentation using synthetic data for time series classification with deep residual networks," 2018, https://arxiv.org/abs/1808.02455.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8849734, 12 pages
https://doi.org/10.1155/2020/8849734

Research Article

Train Type Identification at S&C

Martina Kratochvílová,[1] **Jan Podroužek**,[1] **Jiří Apeltauer**,[2] **Ivan Vukušič**,[3,4] **and Otto Plášek**[3]

[1]*Institute of Computer Aided Engineering and Computer Science, Faculty of Civil Engineering, Brno University of Technology, Veveří 331/95, 602 00 Brno, Czech Republic*
[2]*Institute of Road Structures, Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic*
[3]*Institute of Railway Structures and Constructions, Faculty of Civil Engineering, Brno University of Technology, Brno, Czech Republic*
[4]*Výzkumný Ústav Železniční, a.s. (VUZ), Novodvorská 1698, 142 01 Prague, Czech Republic*

Correspondence should be addressed to Martina Kratochvílová; kratochvilova.m@fce.vutbr.cz

Received 28 August 2020; Revised 19 October 2020; Accepted 6 November 2020; Published 24 November 2020

Academic Editor: Petr Dolezel

The presented paper concerns the development of condition monitoring system for railroad switches and crossings that utilizes vibration data. Successful utilization of such system requires a robust and efficient train type identification. Given the complex and unique dynamical response of any vehicle track interaction, the machine learning was chosen as a suitable tool. For design and validation of the system, real on-site acceleration data were used. The resulting theoretical and practical challenges are discussed.

1. Introduction

A key and irreplaceable part of every railway track is its switches and crossings (S&C). In terms of dynamic effects, these are some of the most loaded track sections. They not only interrupt runway continuity but also see a change in track stiffness. S&C represent only a small part of a railway network in terms of the length of track; however, their maintenance (which includes special rail structures such as road crossings), relative to conventional tracks, can involve high maintenance costs [1–3]. The primary reason for this is the complex force effect that enables the train to pass through the S&C section; another factor is the requirement to maintain the upkeep of the many components that make up the S&C. As well as the significant direct costs, the maintenance of these sections generates indirect costs (due to delayed trains because of maintenance or slower travel, alternative train routes, and even alternative forms of transportation). Therefore, it is essential that any maintenance of such sections should be planned carefully. However, currently, there is no reliable device which can tell us when it is time for maintenance [4]. Moreover, it is a well-accepted fact that structures respond in a very uncertain manner to probabilistically different motion events while there is very limited a priori knowledge on the structural behaviour [5].

For the above reasons, condition monitoring of railways (not only S&C) is a very current topic. In recent years, various sensors and methodologies for measuring and evaluating results have been developed [6–13].

According to [14], machine learning (ML) methods in S&C are often used for condition monitoring and evaluation in data-based fault detection and diagnosis systems. In these cases, they help in large data to search for features that match different failure mechanisms.

This paper is focused on the first part of the self-diagnostic system for railway switches and crossings (S&C)—Train Identification System (TIS). A similar system for predictive maintenance for rail switches is, for example, Konux [15] or ESAH-M [16].

TIS is based on real on-site data from the acceleration sensor and in the future is assumed use of embed vibration

acceleration sensors. Accelerometers provide various benefits including the following temperature stability (over a wide range of temperatures), wide frequency response, linearity, adaptability, and ruggedness. As such, they are suitable for fully operational online measurement in the long term. Different dynamic effects can be observed for each train type [17, 18]. To obtain an accurate comparison of these effects, it is vital that the same train types are compared at the same passing speeds. A precise comparison can help in the detection of faults and/or deterioration at the very early stages. The main benefits of this approach are use of predictive maintenance which can reduce costs [19] and better planning of the regular maintenance and decision support for infrastructure manager about maintenance activity (such as tamping, component replacement, and surface build-up welding).

This research was part of an initiative whose aim is to investigate, develop, validate, and initially integrate radically new concepts for switches and crossings that have the potential to lead to increases in capacity, reliability, and safety while reducing investment and operating costs.

The first part of the article is dedicated to the description of measurement of the data, selection of datasets, their analysis, and building of a vector for machine learning. The second part deals with the application of support vector machine and validation of the results.

2. Dataset

2.1. Measured Data. The used data were collected during several measurement campaigns which took place in the years 2013 and 2014. The measurements were made primarily on two locations: Choceň and Ústí nad Orlicí and on two S&C per each location. The accelerometers were mostly placed around the crossing because of the maximal dynamical effects on rails and bearers during the train passage. The placement of the sensors is shown in Figures 1 and 2.

All data were acquired with measuring system Dewetron DEWE 2502 and acceleration sensors triaxial piezoelectric Brüel &Kjær 4524 B001 (for rail) and piezoelectric Brüel & Kjær 4507 B004 (for bearer). Sampling frequency was set on 10 kHz, high-pass filter frequency 3 Hz, and low-pass filter frequency on 1000 Hz [20].

The train speed was measured by radar speed gun Bushnell.

Acceleration is measured at several points along the crossing. The observed magnitude is chosen as the vertical acceleration of the bearer under the crossing nose, as this is the point at which the greatest dynamic effects on bearers occur. Undoubtedly, any damage to the trackbed or the crossing would influence the frequency response. Figure 3 shows an example of an acceleration plot, which was observed during the passing of a train.

2.2. Measurement Selection. The full dataset consists of over 100 complex measurements (in addition to the acceleration, which was measured at several S&C locations, train speed and rail displacements were also measured), taken from trains passing through crossings at a number of stations. However, for building successful classifier, it is required to have data that were obtained under the same or very similar conditions. In Figure 4 are shown differences in vectors obtained from Ústí nad Orlicí and Choceň. The individual columns (i.e., the corresponding scalars of the individual vectors) were normalized and these normed scalars were assigned a colour shade on a scale between yellow and orange based on the value. The locations have different types of bearers, and therefore, the acceleration signals are incomparable. It is easy to see that first two and second two rows are from distinct locations. Due to the higher number of measurements, the data from Choceň were chosen. Though there are measured signals from two S&C from this location, it was not possible to use them for training of one classifier as each one has different dynamical behaviour due to the distinct conditions of stiffness of its support. Because of all these restrictions, there left very little data suitable for training and testing artificial intelligence (AI). Another complication with data comparability was the renovation of the common crossing that was done between the measurement campaigns and so the latter passages were measured under other conditions. Because of the lack of the training data, it was decided to keep these passages. At the same time, this allowed to verify the robustness of the classifier for this kind of S&C reparation.

2.3. Train Details. The available dataset was able to meet the requirements mentioned for only four trains. However, the number of measurements was still sufficient to build the minimum number of data subsets for training and testing. The mechanical properties of the trains are given in Table 1. The trains are shown in Figure 5.

2.3.1. Locomotive Classes 151, 362, and 380. These locomotives are very similar, in terms of both geometry and design. They were made by Czech industrial conglomerate Škoda Works. All the locomotives are electrical; however, class 151 can be powered only by direct current (3 kV) while both 362 and 380 are adapted for other standardised voltages and current (362 is equipped with double system 3 kV DC/25 kV 50 Hz and 380 is equipped with even triple system 3 kV DC/25 kV 50 Hz/15 kV 16,7 Hz). The maximal speed is 160 km/h for type 151, 140 km/h for 362, and 200 km/h for 380. Locomotives 151 and 380 have the same fixed wheelbase and pivot spacing.

2.3.2. Leo Express. Leo Express train is Stadler Flirt IC five-car electric multiple unit. That means the train signal should always have 12 peaks. The major difference between LE and previously mentioned trains is system of chassis. The LE has two powered bogies (at both ends of the train) and 4 Jacobs bogies [21] between the carriages. These characteristics allow well distinguishing the Leo Express signal from other train types. The maximal travel speed is 160 km/h.

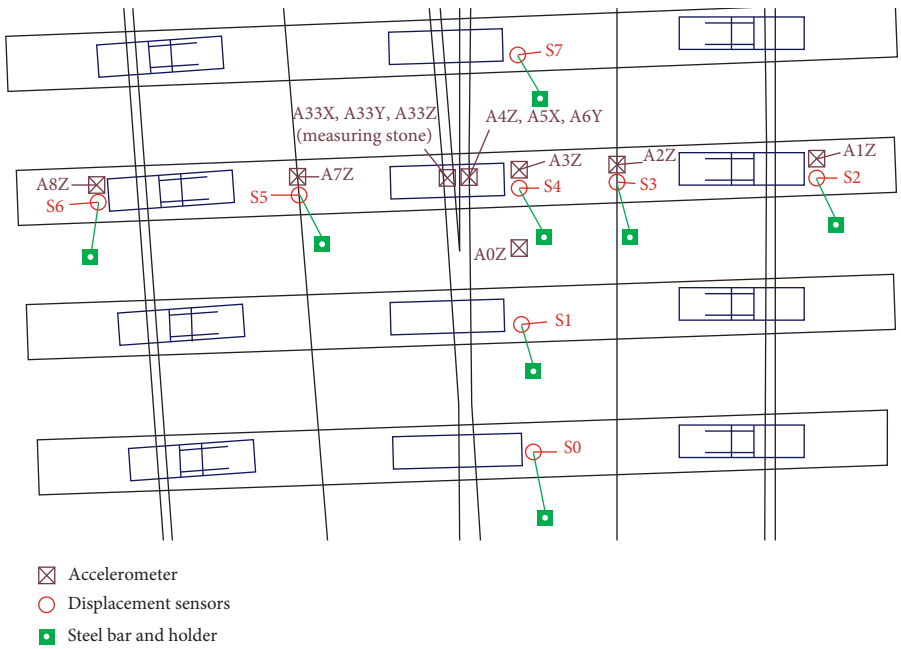


Figure 1: Measurement methodology used for data acquisition from the crossing part of the S&C.

Figure 2: On-site installation of the sensors according to measurement methodology above.

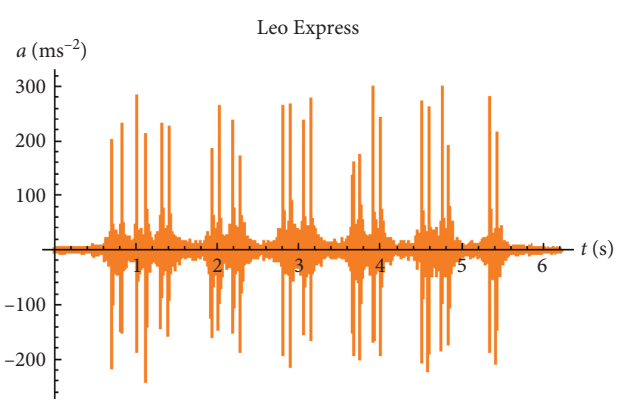


Figure 3: Acceleration plot of the bearer under the crossing nose.

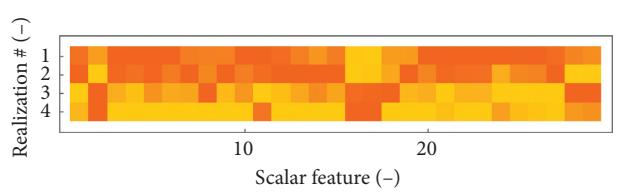


Figure 4: Comparison of data vectors from Ústí nad Orlicí (rows 1 and 2) and Choceň (rows 3 and 4).

3. Data Analysis

There are 3 considered groups of methods: (i) complex time-frequency methods, (ii) methods based on statistical processing, and (iii) combination of the two previously

Table 1: Mechanical properties of the trains.

Locomotive class	151	362	380	Leo
Distance between pivots (m)	8.3	8.3	8.7	16.0
Axle spacing (m)	3.2	3.2	2.5	2.7
Max. axle loading (t)	20.5	21.75	21.5	—
Weight (t)	82.0	87.0	86.0	150.0[1]

[1]Total weight of the whole five-car unit.

(a) (b) (c) (d)

Figure 5: Train types: (a) 151, (b) 362, (c) 380, and (d) Leo Express.

mentioned. The first group analyses signal simultaneously in both time and frequency domains. There are several time-frequency distribution functions, such as wavelet transform (WT), Wigner–Ville transform (WVT), and short-time Fourier transform (STFT). With these methods, it is possible to conduct a sufficiently detailed analysis of the structure's frequency response to reveal minor differences in the individual signals that can suggest that there are vehicle and track faults. However, the major disadvantage of these methods is their significant requirement for data performance and, hence, for computing resources. This is problematic when attempting to ensure long-term in situ measurements for multiple S&C. The use of expensive sensors is also necessary to ensure the high quality of the signals; however, this may not align with other deployment objectives.

The second group of analysis methods can be used as an alternative, and these are based on statistical processing. For example, it is possible, with these methods, to evaluate the maximum amplitudes, as well as their count, standard deviation, and long- and short-term variance. This group of methods is, in essence, the opposite of the time-frequency methods because they have little sensitivity to imperfect input signals, their computational difficulty is negligible (in comparison with the first group of methods), and the device built as a result can be inexpensive. However, the main disadvantage of the second group is that there is limited information in the frequency domain, meaning that the detection of any defects might be too late to be of use. Nonetheless, the time domain of the signal provides very accurate information.

The methods in the third group are a combination of the two approaches mentioned previously, enabling the time domain of the signal to be analysed using statistical methods. In identified areas of interest (for example, maximum amplitude axles), a simple frequency analysis can be conducted using the selected signal subsection's frequency spectrum and its statistical properties. This method is advantageous for our research because it is economical on computer performance while being able to adequately describe the signal.

3.1. Signal Evaluation in the Time Domain Using Statistical Methods. The use of statistical methods was inspired by previous research [22] that focused on train detection and classification. This innovative method evaluates the accelerometer record as a windowed variance of acceleration, based on 12–20 records at a sampling rate of 100 Hz, a sensitivity of ±4 g, and a resolution of 10 bits. Despite the minimalistic resolution (as well as the minimal power and hardware requirements), the system can achieve very accurate results as well as detect and classify trains with over 95% precision. It has a battery capacity of 180 mAh (units of percent of conventional smartphone battery capacity), enabling the device to take measurements for approximately two weeks. An SD card is used to store the results.

As a truly economical system, this can easily be scaled and expanded to other variables (as demonstrated in Figure 6), including maximum number, standard deviation, and absolute and local maximum, requiring minimal power and hardware. To identify short signal sequences, time-based input analysis can be used when a more detailed analysis is conducted in the frequency domain (as shown in Figure 7).

3.2. Signal Evaluation in the Frequency Domain Using Statistical Methods. Evaluation is performed in the frequency domain with the Seewave package [23], using R language to process the model example. Practical deployment would require a lower level of programming language, probably at

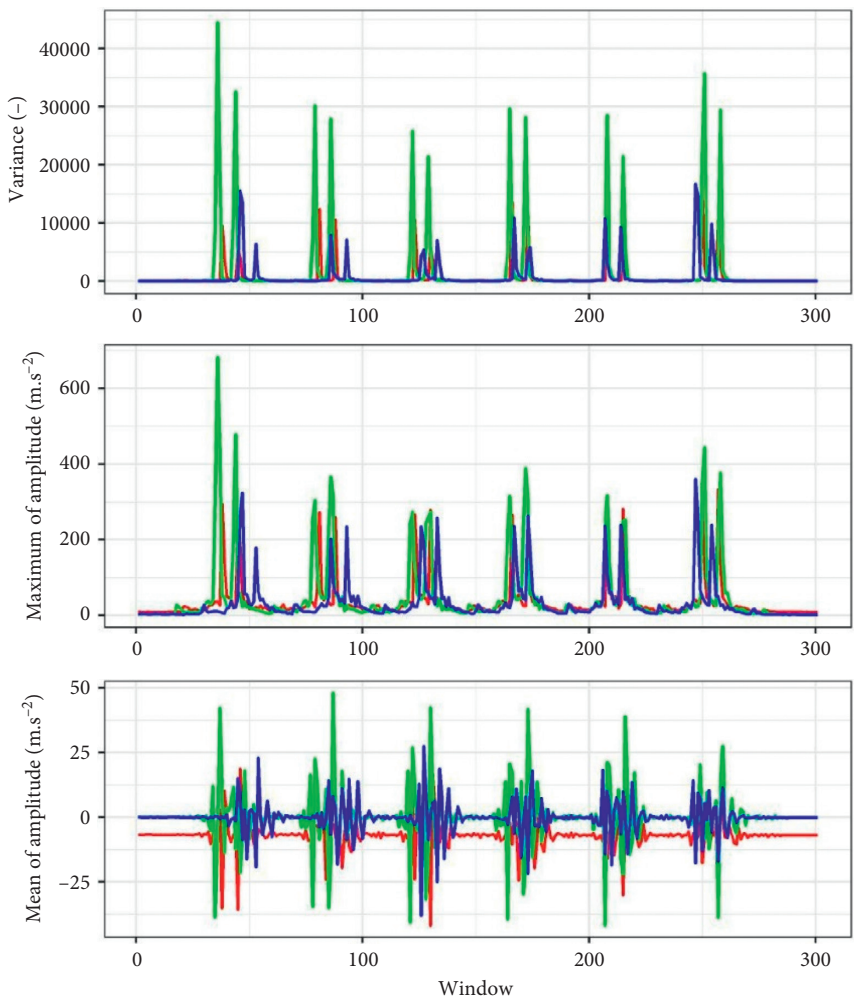


Figure 6: LEO Express train signal windowed variance (three passages). Window size: number of samples/300. Top: windowed variance value. Middle: windowed maximum. Bottom: windowed mean.

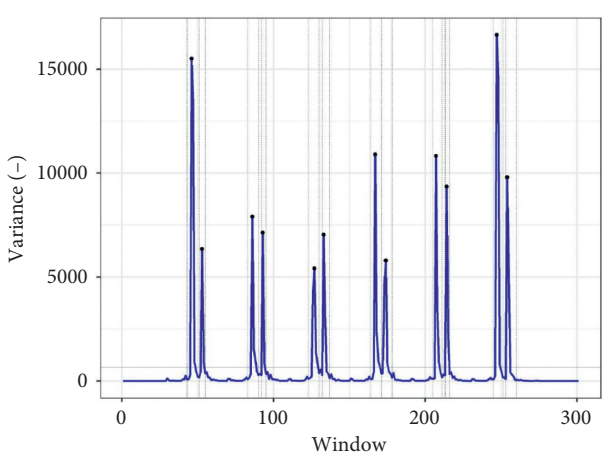


Figure 7: Peak detection of windowed variance. Window size: number of samples/300. Defines the area in the time domain; the analysis is then conducted in the frequency domain.

the firmware level. However, the frequency analysis is very complex, and therefore, only a limited sample of data is performed. Statistical methods are used for sample selection in the time domain. As a result, the processing is computationally efficient, particularly regarding the amount of memory used. Described by a relatively short vector of statistical properties, the spectrum transforms into a discrete probability density (as illustrated in Figure 8).

The analysis is supplemented further by the maximum and minimum frequencies at three density intervals (0.0001–0.00015, 0.00015–0.0002, and 0.0002–0.0004). Combining the scalar features of the statistical properties in the frequency and time domains enables a vector to be obtained that represents the signal in the time-frequency domain but with minimum resources in comparison with traditional methods such as WT or STFT. However, it should be noted that not all vector values are relevant.

3.3. Machine Learning Methods: Building a Vector. A wide variety of data and formats can be used as inputs for ML. A high-resolution accelerometer signal (such as 10 kHz) as an input is likely to be the simplest option. However, this would require a particularly powerful computing subsystem with substantial memory, which would render the method unsuitable for use in situ or on larger scales. In addition, it is

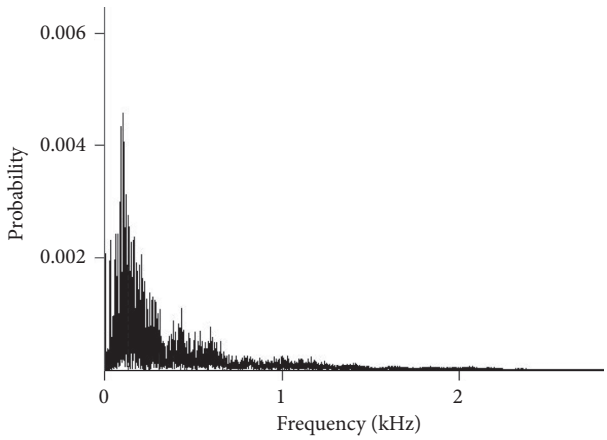


Figure 8: Discrete probability density plots of the Leo Express.

not guaranteed that such a procedure will lead to the best results. Therefore, the selection of the descriptive features is an important step in the ML-based identification process, along with the creation of a sequence of n scalar features (or representations) by reducing the recorded acceleration time history. The representations will include event duration, total amount of vibration caused by the train, number of peaks extracted from the windowed variance, average distance between peaks, maximum peak value, average peak amplitude, average peak area under the curve, total area under the curve, and variance of peak distances. The computational power requirements are reduced by several orders of magnitude by using the combined time-frequency characteristics vector defined in the previous section. However, it is highly likely that some of the features will be random or similar for each individual train, and including such features in the calculation could easily confuse the machine (e.g., SVM or neural network), leading to incorrect results.

The initial set of 27 scalar features contains number of peaks (number of axles), their minimum and maximum, standard deviation, and total sum. Furthermore, the mean of the signal, standard deviation, median, standard error of the mean, 25% and 75% quantile, interquartile range, centroid, skewness, kurtosis, spectral flatness measure, and minimum and maximum frequencies for a given interval of discrete probability density. This vector was reduced to 5 using an iterative optimisation process, whereby accuracy was maximised by the minimisation of training time, evaluation time, and classifier memory and loss. The initial set of 27 scalar features contains number of peaks (number of axles), their minimum and maximum, standard deviation, and total sum. Furthermore, the mean of the signal, standard deviation, median, standard error of the mean, 25% and 75% quantile, interquartile range, centroid, skewness, kurtosis, spectral flatness measure, and minimum and maximum frequencies for a given interval of discrete probability density. The use of the whole vector was considered; however, due to the low number of data and the large number of possible parameters, this is an overdetermined problem, and therefore, according to the authors, it did not make sense to do a detailed sensitivity analysis. Figure 9 shows visualization of velocity and scalar features which were selected for description of the individual train passages. The data are sorted by train type. It can be seen that values of some scalar features of some classes are correlated with the train type, and hence, they are clustering whereas other classes have values widely scattered. For this reason, there is a need to have more than one scalar features to correctly classify the signal. However, as was said earlier, it is not advantageous to use all 27 scalar features not only because of high computational demands but also because of the well-known phenomenon of curse of dimensionality [24]. Velocity was not selected into the vector because it is secondarily included in the other characteristics and for some S&C may be strongly influenced by the position in the track and not by the train type.

The following scalar features were chosen to describe the train passage:

(i) n_{peaks}: number of peaks detected during windowed variance. The R language findpeaks function was used for the detection. The number represents the number of axles on the train.

(ii) $\text{peaks}_{\text{sum}}$: sum of maximum values of npeaks detected. To a certain extent, this expresses the absolute amount of dynamic energy that is transmitted to the sleeper

(iii) sem: the random sampling process is described using the standard error of the mean. The variation in measurements is described using the standard deviation of the sample data. The sem is a probabilistic statement that describes how the sample size, considering the central limit theorem, will provide a better boundary on estimates of the population mean.

(iv) IQR: the interquartile range, which is also known as the midspread or the middle 50% (or, technically, H-spread), is a measure of statistical dispersion, which is equal to the difference between the upper and lower quartiles, or between the 75th and 25th percentiles. The IQR value represents the bandwidth of energy transferred to the sleeper.

(v) prec: the spectrum's frequency precision.

4. Machine Learning-Based Analysis

The aim of this study is to confirm the hypothesis regarding the possibility of using recorded acceleration data to identify specific train types at rail S&C. Utilisation of ML methods [25] seems appropriate due to the unique and complex dynamic interactions involved in the process, including those involving the vehicle itself and the wheel, as well as railway S&C components and ballast, and also the recorded signal's stochastic components. A further consideration is that ML might be able to identify not only a specific train type but also any possible damage to the wheel surface and parts of the S&C [26].

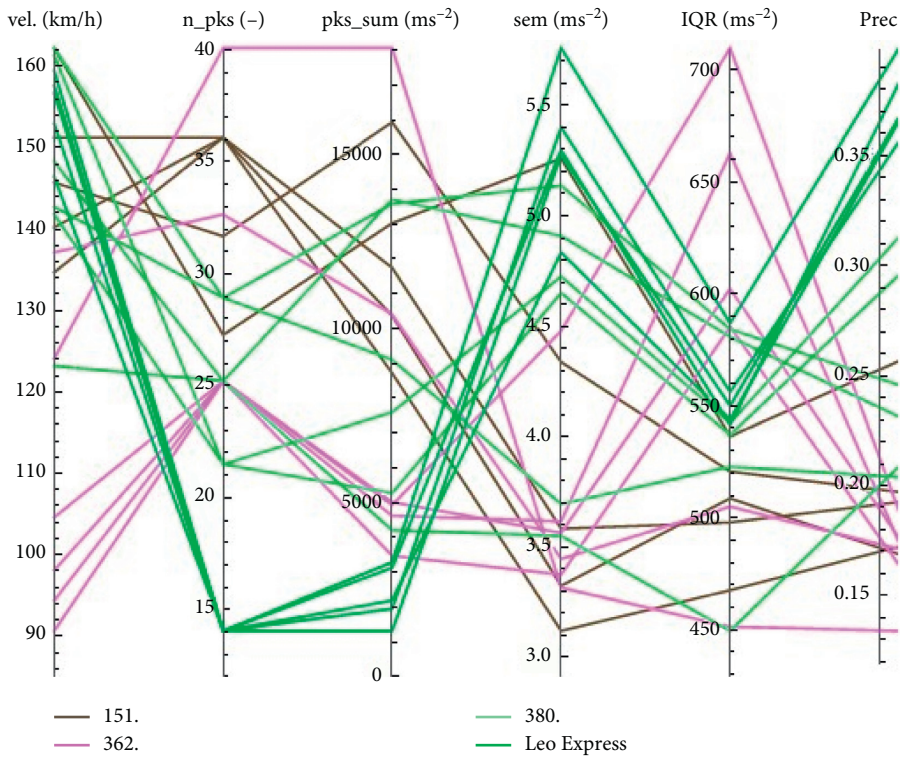


Figure 9: Visualization of clustering of scalar features for individual train passages sorted by train type.

An in-depth literature review showed that the use of measured acoustic or acceleration signals with ML to identify train type was performed successfully on a segment of plain-line railway [22]. However, no record was found of the successful application of ML, genetic algorithms, or pattern recognition [27] for train type identification at S&C.

4.1. Comparison of ML Methods. Currently, there are many machine learning methods that differ in the structure and complexity of the algorithm and the suitability for use with different types and sizes of input data. Based on recommendations derived from the literature review, as well as initial investigations using the available ML methods at Mathematica [28], the support vector machine (SVM) was identified as an optimal classifier. The following methods were considered:

(i) A decision tree [29] is a structure designed as a flowchart. Internal nodes represent "tests" for particular features; branches represent the outcomes of the tests; and the leaves represent classes or value distributions.

(ii) Gradient boosting [30] is an ML technique used for regression and classification problems. It produces an ensemble of trees that represent a prediction model. The trees are trained in sequence with the aim of compensating for the weaknesses of previous trees.

(iii) Logistic regression [31] uses a linear combination of numerical features to model the log probabilities of each class. However, its biggest disadvantage for our task is strong sensitivity for outliners.

(iv) In a Markov model [32], each class has a computed n-gram language model during training. During testing, each class's probability is computed according to Bayes' theorem.

(v) A naive Bayes [33] uses an assumed probabilistic independence of features. This method is convenient for large datasets with high dimensionality because it can identify the most significant features.

(vi) Nearest neighbours [34] use instance-based learning. It is easy to implement and works well for multiclass problem, but as the datasets grow, speed and efficiency of the algorithm decline fast. Another disadvantage is sensitivity for outliners and problems associated with curse of dimensionality.

(vii) The random forest [35] uses ensemble learning for classification and regression. It operates by constructing a number of decision trees. The prediction offered by the forest is obtained using the most common class or the mean value of the tree predictions. The training set is divided such that each decision tree is trained on a random subset of features. This algorithm is easy to train because there are not many options for tuning. When there are large input datasets, random forest gives robust model.

(viii) A neural network (NN) [36] is made up of stacked layers. Each layer performs a simple computation.

Starting from the input layer to the output layer, information is processed one layer at a time. The neural network is trained to minimise the training set's loss function using gradient descent and naturally learns nonlinear decision boundaries; however, it often converges to local minimums and can start to consider noise as a part of pattern and therefore overfit the classifier. The NN is parametric; this means that its size is constant with growing input datasets. There are many setting possibilities and it requires experience to set up the algorithm correctly. For this reason, the NN is not advantageous for TIC, which should be operated by engineers not by scientists.

(ix) Unlike neural network, the support vector machine [37] can produce reliable results even with small input datasets. Moreover, it is not sensitive to outliners. The principle is to find an optimal hyperplane dividing areas of different classes. The word "plane" can be somewhat misleading because it does not always have to be a flat plane (or line in 2D). The SVM is linear in its natural form, but it is possible to use other kernel functions that allow to operate in multidimensional space without calculating data coordinates. This can greatly save computing time. In this classifier was used radial basis function kernel. Another distinction is that SVM is nonparametric, and therefore, its complexity increases with the number of training samples. This means that SVM may be beneficial for this research, where is only small input datasets, but in actual implementation with multiple train type classes with higher number of passages, the calculation may take too long.

In this research, machine learning and its postprocess were performed in Wolfram Mathematica 11.1 [28]. The same analysis with the same inputs was also run in version 11.2 but with worse results. Even the choice of SVM as a best method was not validated in the newer version and gave better results for neural network. This may be caused by distinct setup of the embedded algorithm in both versions.

Comparison of ML methods shows the accuracy, training times, and required computation memory for some of the previously mentioned methods (Table 2). It can be seen that SVM gives the highest accuracy, but training takes twice as long as the second slowest method and even nearly 50 times longer than the fastest. However, it should be noted that the nearest neighbour method is the fastest because it does not need any training time—samples are sorted according to the class of their nearest neighbour (or k-neighbours).

4.2. Support Vector Machines. In terms of implementation, SVMs are regarded as binary classifiers [25]. Features are extracted from the examples using a kernel function. During training, the classifier locates the maximum-margin hyperplane that separates the classes. Then, the problem of multiclass classification is reduced to a set of problems of binary classification (using a strategy of one-versus-one or one-versus-all). The LibSVM framework in C/C++ is used in the implementation.

Although classification using SVM can be controlled in a number of ways [28], such as gamma scaling parameter, kernel type, polynomial degree, and multiclass strategy, the training dataset is characterised reasonably well by the automatic settings. However, the training dataset is somewhat limited in terms of repeated identical observations (i.e., the same train on the same switch at a similar speed), which means that a detailed analysis of the effects of any particular setting is difficult.

Full validation of the classifier is impossible due to the limited number of comparable train passages. In the smallest classes, it is only possible to use one train passage for validation, whereas it is possible to use the remaining four comparable train passages for training. This is the case for all combinations. In total, 19 train passages are used, with the recorded acceleration time history being reduced to 5 scalar features.

4.3. Building of Train and Test Sets. Due the low number of comparable train passage in the classes, the reliability of the classifier highly depends not only on the selection of the scalar features, but also on the choice of the vectors (passages) for the training set. To avoid cherry-picking and decrease possibility of incorrect results due to the inappropriate selection of data for training and testing, the bootstrap analysis was performed. Bootstrapping is a compute-intensive method for statistical data analysis [38]. The train passages for the training subset were chosen randomly for each class and the spare ones were used for testing. That means, as the smallest class has only 5 comparable train passages, the training set has 4 vectors per class and one vector for testing. According to [39], the imbalance in the size of the classes can significantly influence the results. Therefore, all classes for training have the same size of 4 passages. Figure 10 shows visualization of all train passages used for ML. The vectors (each containing 5 scalar features) were projected into two-dimensional space with Mathematica built-in function "DimensionReduce." The class 362 has two outliners which can easily confuse the classifier if selected into training subset or be falsely classified during validation. Furthermore, it can be seen that there is no clear boundary between classes 151 and 380. However, it is possible that, with a larger number of samples, the separation of groups would be more obvious.

4.3.1. Implementation of SVMs. As soon as the sets were ready, the ML was performed and classifier was built. The result of the consecutive testing was confusion matrix. This process of building subsets, training and testing, was repeated 1000 times. As the outcome of this repetition process, 1000 confusion matrices were obtained (i.e., one matrix per one subsets selection).

Table 2: Comparison of ML methods.

Method	SVM	Neural net.	Log. reg.	Nearest neigh.	Rnd. forest
Accuracy (%)	75	58	67	58	50
Train. time (s)	2.0715	0.9397	0.293	0.0406	0.0529
Memory (kB)	323.384	219.512	189.240	126.824	199.520

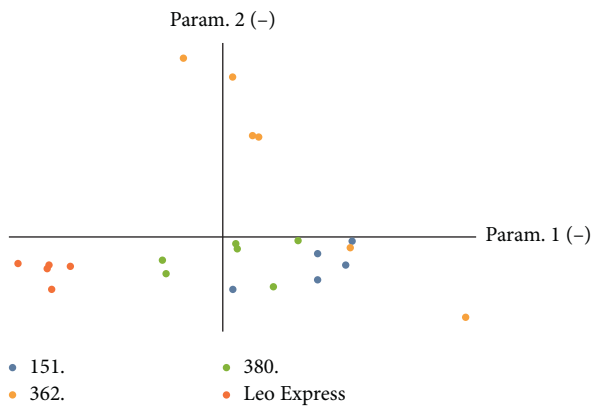


Figure 10: Visualization of clustering of scalar features for individual train passages sorted by train type.

In ML, a confusion matrix (also known as an error matrix [40]) is a very specific table layout that allows the performance of the supervised learning to be visualised (it is most frequently known as a matching matrix in unsupervised learning). Each row of the matrix indicates instances in a predicted class; each column indicates instances in an actual class (or vice versa). The name of the matrix is taken from the fact that it enables the user to check whether the system is confusing (i.e., mislabelling) two classes. It is a particular type of contingency table that has two dimensions (an "actual" and a "predicted" dimension), as well as identical sets of "class" in each dimension (each combination of dimension and class is identified as a variable in the contingency table). There are three random examples of confusion matrices from the analysis in Figure 11.

From all 1,000 matrices, one average matrix was evaluated (i.e., total sum of all results on the same location in matrix was divided by number matrices). For easier understanding, values in each row of the matrix were rescaled to give total sum of 1 so it is possible to seen probability of (miss)classification for this class. Because the test sets were not the same size, the colour of each field tells the information about significance of the testing—the darker, the higher number of test samples. That means that if there are in test subset, for example, 6 train passages from the same train type for testing and it gives the probability of correct classification 0.9, it is more reliable than if there is only 1 testing passage.

The confusion matrix shown in Figure 12 shows a perfect match for the train type Leo Express. This result was expected due to the big differences in train construction (Jacobs bogie). For the locomotive classes 151, 362 and 380, the prediction is worse due to the fact that the trains are very similar (weight, number, and distance of axes) and there was too little data for capturing such subtle differences. The locomotive class 151 is correctly classified in 70% of cases and in 25% of cases is falsely classified as a 380. In the opposite case, class 380 is classified correctly in 61% of cases and confused with 151 in 39% of cases. The classification of class 362 is reliable in 70%.

Although the data contain passages from before and after the common crossing was renovated, the identification method is sufficiently robust, based on the probabilities, to allow for railway crossing component modification, provided that measurements are obtained at the same locations, and as long as the primary objective is TIS only, not condition assessment.

5. Summary and Concluding Remarks

In this paper, the authors have conceptually approached the AI-assisted Train Identification System (TIS), a component of the self-diagnostic system for S&C, utilizing real on-site acceleration data from TEN-T railway lines in Czech Republic. This research is part of the S-CODE project; the overall aim is to investigate, develop, validate, and perform initial integration of radically new concepts for S&C with the potential to increase their capacity, reliability, and safety, while reducing investment and operational costs. Presented approach is unique in attempting the TIS based on measured acceleration time histories in S&C rather than in straight track.

The presented accuracies of the various 5 ML classifiers are clearly limited due to the number of uncontrollable variables and uncertainties, as well as due to limited number of comparable train passages, considering the dimensionality of both the physical problem and the abstract models. As the classification procedure can be sensitive to unequal class sizes, all training classes (train types) have the equal size of 4.

Although a bootstrapping analysis has been performed (1,000 training and testing subsets) in order to fully utilize the experimental evidence and to more objectively select the data for training and testing, the resulting average confusion matrices show prohibitive probabilities, which can be attributed to similarities of the 151 and 380 locomotives, low number of observations, and complex dynamic interactions at S&C in general.

Nevertheless, based on the presented theoretical and practical arguments, it can be concluded that the support vector machines (SVM) can be recommended as most suitable ML method. This conclusion is in line with the published evidence (TIS based on straight track measurements) and is supported by the presented comparison of alternative ML methods. The obvious trade-off for highest accuracy, the increased training time, and memory,

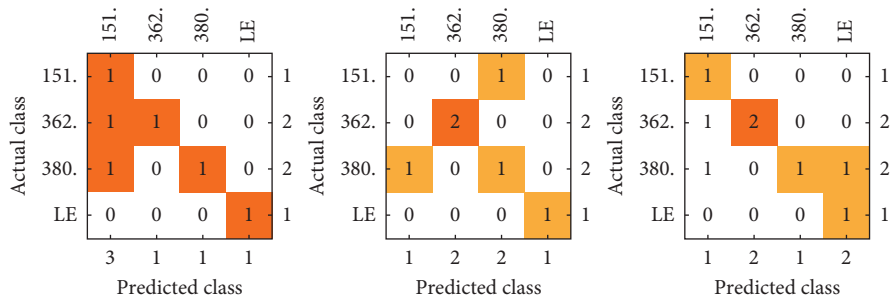


Figure 11: The 3 random confusion matrices obtained by random selection of train and test sets.

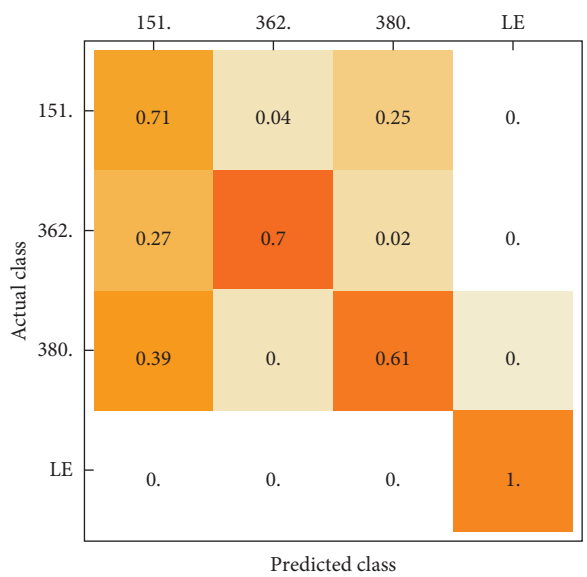


Figure 12: An average confusion matrix calculated during cross-validation using the SVM classifier. The accuracy is relatively poor at 61%, but this can be attributed to the lack of training data and train similarities.

however, is relatively cheap considering the efficiency and availability of current low (energy harvested battery powered) powered computer modules and relative to hardware resources required for statistical preprocessing of the recorded vibration time histories.

In fact, the average accuracy of 75% for SVM-based TIS at S&C cannot be considered entirely off if the published results from straight track TIS using SVM yields accuracy of 96%, and considering the inherently more complex and uncertain response of S&C compared to straight track and the clear similarities of the 151 and 380 locomotives.

For future applications within the system of early warning, it would be advisable to implement the SVM method, use it within the experimental envelope, avoid excessive extrapolation (as can be generally recommended for all ML methods), and combine the diagnostic from S&C with straight track measurement, where it is possible to identify defects on carriage, such as a flat wheel. This would dramatically improve the sensitivity and specificity of the TIS by, e.g., avoiding false positives from boogie defects. Although current optical systems can be used to identify trains by detecting and evaluating the mark placed on each locomotive, these systems are relatively expensive and sensitive to maintenance and weather conditions, compared to the vibration data-driven ML models.

One of the contributing factors to the overall uncertainty is the variable number of passengers in each wagon, significantly affecting the dynamic characteristics of the train formation. This particular aspect can be approached by trimming the signal so that only the locomotive remains, resulting in easier-to-classify data while simultaneously reducing hardware requirements. However, it would be necessary to define objective and universal applicable method of trimming, due to the complex interference of the vibrations caused by the locomotive and the following car, the nonuniform number of locomotive axles, or the presence of Jacobs bogie. In addition, by shortening the signal, some data that can provide valuable information about the condition are lost, and, most importantly, for evaluating the locomotive-only signal, analytical approaches are typically sufficient (classification based on, e.g., distance and number of axles), i.e., ML methods are not required at all.

From a pure TIS perspective, best input would clearly be represented by repeated passages of (specially scheduled) separate locomotives; however, such system could hardly be considered as an early warning system, but a preventive monitoring, as is routinely done, e.g., in the field of structural health monitoring of bridges with scheduled passages of specialized instrumented vehicles.

Although the cross-validation options available clearly limit the statistical significance, the results are unique in demonstrating that

(i) ML- (SVM-) based TIS at S&C is feasible if, within the S&C, the monitoring location is consistent. In cases in which the monitoring location is not consistent, identification is not successful.

(ii) Specifically the approach using SVM is insensitive to common crossing renovation, i.e., data from before and after the renovation can be combined, if only TIS without S&C condition assessment is considered.

(iii) The input vector that reduces full recorded time histories to a set of scalar characteristics must always be chosen subjectively so that it characterises all important features sufficiently while maintaining realistic hardware requirements stemming from the

intended in-situ implementation on energy harvested battery powered modules.

(iv) During an iterative optimisation process in which accuracy is maximised and training time, evaluation time, and classifier memory and loss are minimised, the initial vector of 27 scalar features is reduced to 5.

Abbreviations

AI: Artificial intelligence
S&C: Switches and crossings
ML: Machine learning
TIS: Train Identification System
NN: Neuron network
SVM: Support vector machine
LE0: Leo Express train
WT: Wavelet transform
WVT: Wigner–Ville transform
STFT: Short-time Fourier transform.

Data Availability

All data are available on request through corresponding authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is part of the S-CODE project which received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 730849. The support of the FAST-J-19-6062 project is also acknowledged. Furthermore, the research was supported by the project TAČR CK01000091Výhybka 4.0.

References

[1] A. M. Zarembski, "Factors involved in turnout maintenance," *Railway Track & Structures*, vol. 3, pp. 12-13, 1995.

[2] E. Kassa and J. C. O. Nielsen, "Dynamic interaction between train and railway turnout: full-scale field test and validation of simulation models," *Vehicle System Dynamics*, vol. 46, no. 1, pp. 521–534, 2008.

[3] W. J. Zwanenburg, "Modelling degradation processes of switches & crossings for maintenance & renewal planning on the Swiss railway network," Technical report, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2009.

[4] C. Ngamkhanong, S. Kaewunruen, and B. Costa, "State-of-the-art review of railway track resilience monitoring," *Infrastructures*, vol. 3, no. 1, p. 3, 2018.

[5] J. Podroužek, C. Bucher, and G. Deodatis, "Identification of critical samples of stochastic processes towards feasible structural reliability applications," *Structural Safety*, vol. 47, no. 2, pp. 39–47, 2015.

[6] M. Sysyn, O. Nabochenko, U. Gerber, V. Kovalchuk, and O. Petrenko, "Common crossing condition monitoring with on board inertial measurements," *Acta Polytechnica*, vol. 59, no. 4, pp. 422–433, 2019.

[7] R. Skrypnyk, U. Ossberger, B. A. Pålsson, M. Ekh, and J. C. O. Nielsen, "Long-term rail profile damage in a railway crossing: field measurements and numerical simulations," *Wear*, vol. 2020, Article ID 203331, 2020.

[8] A. Kowalska-Koczwara, F. Pachla, P. Stecz et al., "Vibration-based damage identification and condition monitoring of metro trains: warsaw Metro case study," *Shock and Vibration*, vol. 2018, Article ID 8475684, 14 pages, 2018.

[9] M. Hamadache, S. Dutta, R. Ambur, O. Olaby, E. Stewart, and R. Dixon, "Residual-based fault detection method: application to railway switch & crossing (S&C) system," in *Proceedings of the 2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1228–1233, IEEE, Jeju, Republic of Korea, October 2019.

[10] G. Jing, M. Siahkouhi, K. Qian, and S. Wang, "Development of a field condition monitoring system in high speed railway turnout," *Measurement*, vol. 169, p. 108358, 2020.

[11] K. Gundampati, *Wireless sensor network (WSN) platform for railway condition monitoring*, PhD thesis, University of Huddersfield, Huddersfield, UK, 2019.

[12] A. Paixão, E. Fortunato, and R. Calçada, "Smartphone's sensing capabilities for on-board railway track monitoring: structural performance and geometrical degradation assessment," *Advances in Civil Engineering*, vol. 2019, Article ID 1729153, 13 pages, 2019.

[13] J. Li and H. Shi, "Rail corrugation detection of high-speed railway using wheel dynamic responses," *Shock and Vibration*, vol. 2019, Article ID 2695647, 12 pages, 2019.

[14] M. Hamadache, S. Dutta, O. Olaby, R. Ambur, E. Stewart, and R. Dixon, "On the fault detection and diagnosis of railway switch and crossing systems: an overview," *Applied Sciences*, vol. 9, no. 23, p. 5129, 2019.

[15] T. Böhm and N. Weiß, "Weichenanalytik—smarte sensoren und künstliche Intelligenz für die rundum gesunde Weiche," *Eisenbahntechnische Rundschau ETR*, vol. 5, pp. 42–45, 2017.

[16] A. Zoll, U. Gerber, and W. Fengler, "Das Messsystem ESAH-M (the measuring system ESAH-M)," *EI-Eisenbahningenieur Kalender*, vol. 1, pp. 49–62, 2016.

[17] S. Iwnicki, *Handbook of Railway Vehicle Dynamics*, CRC Press, Boca Raton, FL, USA, 1st edition, 2006.

[18] I. Vukušič, D. Sadleková, J. Smutný, L. Pazdera, V. Tomandl, and J. Hajniš, "Measurement and analysis of the dynamic effects on the crossings," in *Proceedings of the 3rd International Conference on Road and Rail Infrastructure*, Split, Croatia, April 2014.

[19] V. Reddy, G. Chattopadhyay, P.-O. Larsson-Kraik, and T. Allahmanli, "Technical vs. economical decisions: a case study on preventive rail grinding," in *Proceedings of the 5th Asia-Pacific Industrial Engineering and Management Systems Conference & The Seventh Asia-Pacific Division Meeting of the International Foundation of Production Research*, Gold Coast, Australia, December 2004.

[20] I. Vukušič, *Analysis of the Dynamic Effects in the Turnout*, Brno University of Technology, Faculty of Civil Engeneering, Institute of Railway Structures and Constructions, Brno, Czech Republic, 2015.

[21] Bogies, 2019, http://www.railway-technical.com.

[22] E. Berlin and K. van Laerhoven, "Sensor networks for railway monitoring: detecting trains from their distributed vibration footprints," in *Proceedings of the 2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 80–87, IEEE, Cambridge, MA, USA, May 2013.

[23] Sueur, J. Package Seewave, 2018.
[24] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.
[25] T. M. Mitchell, J. G. Carbonell, and R. S. Michalski, *Machine Learning: A Guide to Current Research*, Vol. 12, Springer Science & Business Media, Berlin, Germany, 1986.
[26] H. Tsunashima, "Condition monitoring of railway tracks from car-body vibration using a machine learning technique," *Applied Sciences*, vol. 9, no. 13, p. 2734, 2019.
[27] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2/3, pp. 95–99, 1988.
[28] Wolfram Research, I. Mathematica.
[29] J. R. Quinlan, "Learning decision tree classifiers," *ACM Computing Surveys*, vol. 28, no. 1, pp. 71-72, 1996.
[30] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.
[31] D. T. Bui, T. A. Tuan, H. Klempe, B. Pradhan, and I. Revhaug, "Spatial prediction models for shallow landslide hazards: a comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree," *Landslides*, vol. 13, no. 2, pp. 361–378, 2016.
[32] K. Seymore, A. Mccallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*, pp. 37–42, Orlando, FL, USA, July 1999.
[33] I. Rish, "An empirical study of the naive Bayes classifier," in *Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, pp. 41–46, Seattle, WA, USA, August 2001.
[34] Y. Liao and V. R. Vemuri, "Use of *K*-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.
[35] Powerful Guide to Learn Random Forest in R and Python, 2015, http://www.analyticsvidhya.com/blog/.
[36] The scuffle between two algorithms -neural network vs. Support vector machine, 2018, http://www.analyticsvidhya.com/blog/.
[37] Understanding Support Vector Machines Algorithm (Along with Code), 2017, http://www.analyticsvidhya.com/blog/.
[38] A. R. Henderson, "The bootstrap: a technique for data-driven statistics. Using computer-intensive analyses to explore experimental data," *Clinica Chimica Acta*, vol. 359, no. 1-2, pp. 1–26, 2005.
[39] F. Provost, "Machine learning from imbalanced data sets 101,"vol. 68, pp. 1–3, in *Proceedings of the AAAI'2000 Workshop on Imbalanced Data Sets*, vol. 68, pp. 1–3, AAAI Press, Austin, TX, USA, August 2000.
[40] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8848149, 14 pages
https://doi.org/10.1155/2020/8848149

Research Article

Prediction on Peak Values of Carbon Dioxide Emissions from the Chinese Transportation Industry Based on the SVR Model and Scenario Analysis

Changzheng Zhu, Meng Wang, and Wenbo Du

School of Modern Post, Xi'an University of Posts and Telecommunications, Xian 710061, China

Correspondence should be addressed to Changzheng Zhu; zhuchangzheng@xupt.edu.cn

Received 25 June 2020; Revised 28 August 2020; Accepted 25 September 2020; Published 23 October 2020

Academic Editor: Petr Dolezel

As the largest emitter of greenhouse gases in the world, the peak values of Chinese CO_2 emissions have attracted extensive attention at home and abroad. The carbon dioxide emissions of the Chinese transportation industry, accounting for 9.5% of total carbon dioxide emissions, is one of the high-emission industries, and its total carbon dioxide emissions continue to rise. Therefore, the accurate prediction of the peak values of carbon dioxide emissions from the Chinese transportation industry is helpful for China to formulate a reasonable policy of carbon dioxide emissions control. This paper, firstly, selects six major factors affecting the carbon dioxide emissions of the Chinese transportation industry. They are the Gross Domestic Product (GDP), population, urbanization rate, energy consumption structure, energy intensity, and industrial structure. Then, it builds a prediction model of carbon dioxide emissions based on Support Vector Regression (SVR). Finally, it analyses the sensitivity of each factor. The predicted results show that, under the baseline scenario, they will reach a peak of 1365.71 million tons in 2040; under the low-carbon scenario, the carbon dioxide emissions of Chinese transportation will peak at 1115.43 million tons in 2036; and in the high-carbon scenario, the peak value will occur in 2046 and the carbon dioxide emissions will be 1738.18 million tons. In order to promote the early peak of carbon dioxide emissions from the transportation industry, it is, firstly, necessary to change the mode of economic growth and appropriately reduce the speed of economic development. Secondly, the energy intensity of the transportation industry is reduced and the utilization rate of clean energy is improved. Thirdly, the industrial structure is optimized. Fourthly, the carbon dioxide emissions of the transportation industry caused by the increased urbanization rate are reasonably controlled.

1. Introduction

With the development of the world economy, the situation of carbon dioxide emissions control is becoming increasingly serious. In November 2019, the United Nations Environment Programme (UNEP) released the Emissions Gap Report, which pointed out that human efforts to control carbon dioxide emissions have been too weak in the recent ten years, and the global carbon dioxide emissions have been always on the rise, leading to broader and more destructive climate influences. In December 2019, the 25th conference of the parties to the United Nations convention on climate change was held in Spain, and the United Nations secretary-general Antonio Guterres has called on all countries to adopt more effective measures to control the growth of carbon dioxide emissions. According to the report from the Netherlands Environmental Assessment Agency, China surpassed the United States as the world's largest country of CO_2 emissions in 2006. Also, according to the International Energy Agency (IEA) data, Chinese CO_2 emissions reached 9.30 billion tons in 2017, accounting for 28.33% of the world's total volumes. Among them, the CO_2 emissions of transportation industry accounted for 9.5% of Chinese total volumes, making it a major carbon emitter in the national economy.

However, compared with other industries, the Chinese transportation industry is not ideal enough in carbon dioxide emissions control because of its multiple and complicated emission sources. Therefore, the prediction of the future peak values of carbon dioxide emissions in the Chinese transportation industry is helpful for the government administration departments to realize the grim situation of carbon dioxide emissions control in the transportation sector, so as to speed up the formulation of more stringent policies for carbon dioxide emissions control.

At present, the relevant research studies of domestic and foreign scholars mainly focus on the influencing factors and prediction of carbon dioxide emissions in the transportation industry. In terms of influencing factors, the research results of most scholars show that the economic level [1–5], population [2, 3, 5, 6], energy intensity [2, 3, 7, 8], energy consumption structure [4–7], urbanization rate [5], and industrial structure [9–11] are the main factors affecting the carbon dioxide emissions of the transportation industry. There are also a few scholars who believe that transportation demand [4], the level of transportation development [6], the added value of the transportation industry [7], transportation intensity [8], the market concentration level [12], energy efficiency [13], average driving distance, and the number of motor vehicles [14] are also important factors affecting carbon dioxide emissions.

As for the prediction of carbon dioxide emissions, the most widely used prediction models include the IPAT model [15], STIRPAT model [16], scenario analysis method [17, 18], and regression analysis method [19]. In the early 1970s, Ehrlich et al. established the famous IPAT equation [20] to study the impact of population on environmental change. However, IPAT equation has a certain limitation, which is to analyze the influence of a changed factor on environmental change on the premise of keeping other factors unchanged, so as to obtain the result of equal proportional influence on dependent variables. In order to solve this limitation, Dietz et al. proposed the random model of environmental impact, namely, the STIRPAT model [21]. However, the STIRPAT model mostly specifies different models by simply adding or deleting variables [22]. Even with the improved STIRPAT model, most of the influencing factors are randomly selected to conform to the multiplication rules of the model. Without necessary theoretical support, the credibility of the empirical results will decline [23]. The scenario analysis method is often used in combination with other methods in practical research centers because it only establishes a set of framework and analysis of environmental impact in each scenario must also rely on other more specific methods. As for the regression analysis method, due to the strict assumption of its equation, it is necessary to know all explanatory variables that cause the change of dependent variables; otherwise, it is easy to have problems such as false regression, resulting in the failure of the hypothesis test. But, there are many influencial factors of carbon dioxide emissions, so it is relatively difficult in this choice.

In the recent years, machine learning methods have been widely applied to the prediction of carbon dioxide emissions. Chen et al. used the artificial neural network (ANN) to predict CO_2 emissions and estimated CO_2 emissions from global reservoirs [24]. However, due to the very slow convergence speed of the neural network algorithm, it is easy to fall into the local minimum [25]. Support vector machine (SVM) is a new machine learning algorithm based on statistical learning theory. Because of its good learning performance, it has been used for classification and regression problems. This solves the defects of the prior method and becomes an effective method of carbon emission prediction [26]. Chen et al. established a prediction model for regional carbon emissions based on support vector regression machine to predict the carbon emissions of Beijing's transportation industry [27]. Song et al. predicted Chinese carbon emissions from 2010 to 2015 based on taking the data of Chinese carbon emissions and influencing factors from 1980 to 2009 as samples and combining with the 12th Five-Year Plan [28]. Xue et al. analyzed the advantages of support vector regression machine model in carbon emission prediction and built a prediction model of carbon emissions based on this. By using the data of carbon emissions in Hebei Province from 1990 to 2015 and its influencing factors, we predicted the carbon emissions of Hebei Province from 2016 to 2015 and provided suggestions for carbon emissions reduction [29].

This paper combines the Support Vector Regression (SVR) machine model and the scenario analysis method, uses their advantages to solve the problem of small sample and nonlinearity to forecast the peak value of the Chinese transportation industry in the three scenarios of high-carbon scenario, benchmark scenario, and low-carbon scenario, and provides references for the government making carbon emission control policies.

2. Establishment of the Prediction Model and Selection of Influencing Factors

2.1. Model Establishment

2.1.1. Selection of the Prediction Model. The carbon dioxide emissions of transportation industry are often affected by economic, social, and other factors. Through comparative analyses, this paper selects the SVR model as the prediction model for carbon dioxide emissions of transportation industry. Firstly, the main idea of SVR is to maximize classification boundaries and adapt to various nonlinear situations by selecting a kernel function, which is more suitable for nonlinear data regression prediction than the traditional prediction model. Secondly, the SVR model can realize efficient transformation from training the sample set to the prediction sample set through the small sample learning method, which can solve the problem of small sample data. Finally, in the perspective of obtaining the global optimal solution, the SVR model will be transformed into a convex optimization problem in the final calculation to ensure the global optimal result. Therefore, the SVR model is selected for carbon dioxide emissions' prediction in this paper [30].

2.1.2. Specific Steps of Predictions. SVR is a prediction method based on structural risk minimization, which can comprehensively consider the fitness and complexity of training samples and achieve the optimal effect in function approximation, regression prediction, and other aspects. In this paper, the nonlinear SVR model is selected for predictions.

The data of carbon dioxide emissions were obtained from transportation industry and related influencing factors for 45 years from 1973 to 2017, and the SVR model is built for carbon dioxide emissions of transportation industry; the specific steps are as follows:

Step 1. Independent variables and dependent variables in the sample data are normalized, so that all data are between [0, 1]. After normalization, all indexes are in the same order of magnitude, which is convenient for comprehensive comparisons and improves calculation accuracy. The normalization method is shown as follows:

$$x_i^* = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \quad y_i^* = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}}. \tag{1}$$

In the formulas, x_i, y_i are elements of the data to be normalized; $x_{\max}, y_{\max}$ are the largest elements in data matrix; $x_{\min}, y_{\min}$ are the smallest elements in data matrix; and x_i^*, y_i^* are normalized data.

Step 2. Kernel function and parameter selection

Firstly, the selection of the SVM kernel function plays a crucial role in the performance of SVM. By means of the LIBSVM toolbox, this paper selects the radial basis kernel function $K(x_i, x) = (\Phi(x_i) \cdot \Phi(x)) = \exp(-\gamma \|x_i - x\|^2)$ as the kernel function of the ε-SVR model after systematic analysis of the selected data of each year. Secondly, after LIBSVM adopts the kernel function, the model parameters that are need to be determined are only penalty factor c and kernel function parameter g. The initial value ε is set, the values of parameters c and g are determined with the grid algorithm, and finally, the optimal parameters are obtained by comparisons.

Step 3. The prediction model is established, simulation is carried out for sample set data by MATLAB, and the deviation degree between the prediction data of the training sample and the actual data is compared. The learning and promotion ability of the SVR model are judged by fitting the prediction index, namely, Mean Squared Error (MSE). At the same time, the data of the test set are substituted into, and their mean square error (MSE) is verified. If it is in line with the expectation, the following prediction analysis is carried out. Otherwise, we go back for continuing learning.

Step 4. The predicted values of influencing factors of transportation carbon dioxide emissions are normalized, they are substituted into the established prediction model, the results are output, and the inverse normalization is carried out, so as to obtain the required prediction data of carbon dioxide emissions under different scenarios. Inverse normalization is an inverse process of data normalization. The formula is

$$y_i = y_i(y_{\max} - y_{\min}) + y_{\min}. \tag{2}$$

2.2. Selection of Influencing Factors for Carbon Dioxide Emissions. From the abovementioned literature studies, the factors affecting carbon dioxide emissions of Chinese transportation industry mainly include the GDP, population, urbanization rate, transportation development level, transportation energy intensity, energy consumption structure, and industrial structure. The level of traffic development is a comprehensive indicator, and there exist some differences in how to measure it quantitatively; therefore, this paper excludes this indicator. In addition, the classical regression analysis requires the independent variables to be linearly independent while the support vector machine (SVM) model does not require the independent variables to be linearly independent. Therefore, when using the regression model of SVM to predict the peak of carbon dioxide emissions, this paper selects six indicators of GDP, population, urbanization rate, energy consumption structure, energy intensity, and industrial structure as the main factors affecting the carbon dioxide emissions of the transportation industry. Among them, the energy consumption structure refers to the proportion of fossil energy consumptions of the transportation industry in the total energy consumptions, and the industrial scale refers to the proportion of the secondary industry in GDP.

2.3. Setting of the Predicted Scenario. If China wants to make the policies of carbon dioxide emissions control suitable for its national conditions, it is crucial to accurately predict the peak values of carbon dioxide emissions. Since the peak values of carbon dioxide emissions vary under different scenarios, scenario analysis is needed to predict the peak of carbon dioxide emissions in the Chinese transportation industry. Different scenarios refer to the different change rates of the six factors in the future. In this paper, three scenarios are set up, namely, the benchmark scenario, the high-carbon scenario, and the low-carbon scenario. They are specified as follows: (1) the baseline scenario: in this scenario, the future change rates of the six influencing factors continue the previous change trends, and the change range is moderate; (2) the high-carbon scenario: in this scenario, the future changes of the six influencing factors will lead to higher carbon dioxide emissions than those under the benchmark scenario, such as the faster growth of GDP and urbanization rate; (3) the low-carbon scenario: in this scenario, the future changes of the six influencing factors contribute to lower carbon dioxide emissions than those under the baseline scenario. The detailed parameter setting of each scenario is given in the following.

3. Data Source and Preprocessing

3.1. Data Sources

3.1.1. Independent Variable. In this paper, the population and urbanization rate come from World Development Indicators (WDI), and the energy consumption structure comes from the International Energy Agency (IEA). Here is the percentage of fossil energy consumption in transportation industry accounting for the total energy consumptions, and the rest of the data come from the comprehensive and publicly published Chinese Statistical Yearbook of past years. The sample interval is from 1973 to 2017.

The energy intensity of transportation industry is expressed by the total energy consumption of the unit conversion turnover, which can measure the comprehensive energy utilization efficiency of the industry. Total energy consumption in the Chinese transportation sector comes from the website of IEA. When calculating conversion turnover, passenger turnover is multiplied by the passenger-cargo conversion coefficient and added to cargo turnover to get the total conversion turnover. The turnover coefficients of passenger-cargo conversion for the four modes of transportation are shown in Table 1. The data of turnover of each transportation mode are derived from the statistical yearbook of China over the years. The converted turnover of the Chinese transportation industry is shown in Table 2. Finally, the energy intensity of the Chinese transportation industry over the years can be obtained by dividing the total energy consumption by the conversion turnover.

The final analyses of the factors from 1973 to 2017 are shown in Table 3.

Table 1: Conversion factor of traffic turnover in China.

Mode of transportation	Road	Railway	Waterway	Airport
Conversion factor (t·km·(p·km)$^{-1}$)	1/10	1	1	1/13

Note: t·km = Tonne·kilometer; p·km = people·kilometer.

Table 2: Converted turnover of the Chinese transportation industry from 1973 to 2017.

Year	Conversion turnover (10^8 t·km)
1973	7314.38
1974	7362.46
1975	8378.88
1976	7996.79
1977	9136.04
1978	11077.59
1979	12778.34
1980	13614.13
1981	13840.80
1982	14869.30
1983	16099.98
1984	18033.79
1985	20882.01
1986	23085.12
1987	25449.03
1988	27490.33
1989	29016.77
1990	29264.84
1991	31302.15
1992	32918.72
1993	34733.32
1994	37719.44
1995	40139.12
1996	40646.43
1997	42738.74
1998	42638.28
1999	45496.81
2000	49693.86
2001	53371.48
2002	56615.42
2003	59577.33
2004	76235.64
2005	87474.71
2006	96730.72
2007	110078.42
2008	119607.96
2009	131692.00
2010	152484.47
2011	171035.03
2012	185927.63
2013	180238.12
2014	194570.60
2015	192023.74
2016	200948.59
2017	212615.34

3.1.2. Carbon Dioxide Emissions Calculation Method. Carbon emissions refer to the general term of greenhouse gas emissions, mainly including carbon dioxide, nitrous oxide, and methane. Among them, CO_2 is the major greenhouse gas that induces the global warming. Since there are no comprehensive statistics for global carbon dioxide emissions at present, most scholars adopt the method of carbon dioxide emissions coefficient, proposed by the Intergovernmental Panel on Climate Change (IPCC) [31], to calculate carbon dioxide emissions through the data of energy consumptions. This method, proposed by the IPCC in 1996, states that the total amount of carbon dioxide emissions is equal to the product of the activity data affecting carbon dioxide emissions and the carbon dioxide emissions coefficient per unit. Therefore, the specific expression formula of carbon dioxide emissions adopted in this paper is as follows:

$$C = \sum_i E_i \times \delta_i = \sum_i E_i \times V_i \times R_i \times F_i \times \frac{44}{12}, \quad (3)$$

where C represents CO_2 emissions from transportation industry; i indicates categories of fossil fuels, that is, the IEA database divides the fuels consumed by transportation into five categories of coal, petroleum products, biomass energy, natural gas, and electricity; E_i represents the energy consumption of fossil fuel i; δ_i is the CO_2 emissions coefficient of carbon energy i; V_i is the average low calorific value of energy i; F_i is the carbon dioxide emissions coefficient of energy i; R_i is the carbon oxidation factor, that is, the carbon

Table 3: Data of traffic carbon dioxide emissions and their influencing factors.

Year	GDP (10^9 yuan)	Population (million people)	Urbanization rate (%)	Energy consumption structure (%)	Energy intensity ($t/10^6$ t·km)	The proportion of the secondary industry (%)
1973	2756.20	881.94	17.18	100	3.12	42.80
1974	2764.13	900.35	17.29	100	3.49	42.40
1975	2733.36	916.40	17.40	100	3.24	45.40
1976	2731.29	930.69	17.46	100	3.57	45.00
1977	2760.39	943.46	17.52	100	3.44	46.70
1978	2797.23	956.17	17.90	100	2.84	47.70
1979	2897.74	969.01	18.62	100	2.57	47.00
1980	3007.38	981.24	19.36	100	2.20	48.10
1981	3078.63	993.89	20.12	100	2.17	46.00
1982	3145.57	1008.63	20.90	100	2.11	44.60
1983	3109.54	1023.31	21.55	100	2.13	44.20
1984	3263.05	1036.83	22.20	100	1.98	42.90
1985	3597.14	1051.04	22.87	100	1.78	42.70
1986	3766.86	1066.79	23.56	100	1.79	43.50
1987	3956.78	1084.04	24.26	100	1.74	43.30
1988	4436.76	1101.63	24.97	100	1.66	43.50
1989	4818.71	1118.65	25.70	100	1.67	42.50
1990	5094.93	1135.19	26.44	96.77	1.51	41.00
1991	5435.16	1150.78	27.31	97.06	1.55	41.50
1992	5881.59	1164.97	28.20	97.30	1.61	43.10
1993	6773.79	1178.44	29.10	97.62	1.73	46.20
1994	8173.02	1191.84	30.02	97.50	1.52	46.20
1995	9286.06	1204.86	30.96	97.67	1.53	46.80
1996	9892.30	1217.55	31.92	98.25	2.00	47.10
1997	10055.60	1230.08	32.88	98.00	1.67	47.10
1998	9969.33	1241.94	33.87	97.92	1.61	45.80
1999	9839.91	1252.74	34.87	98.15	1.69	45.40
2000	10041.96	1262.65	35.88	98.81	2.41	45.50
2001	10250.91	1271.85	37.09	98.82	2.27	44.80
2002	10315.81	1280.40	38.43	98.91	2.32	44.50
2003	10588.00	1288.40	39.78	97.17	2.54	45.60
2004	11325.49	1296.08	41.14	97.60	2.34	45.90
2005	11767.03	1303.72	42.52	97.01	2.19	47.00
2006	12231.34	1311.02	43.87	95.95	2.18	47.60
2007	13182.79	1317.89	45.20	96.23	2.06	46.90
2008	14204.05	1324.66	46.54	94.25	2.08	47.00
2009	14174.12	1331.26	47.88	92.74	1.94	46.00
2010	15154.41	1337.71	49.23	92.89	1.84	46.50
2011	16370.89	1344.13	50.51	92.13	1.80	46.50
2012	16746.90	1350.70	51.77	92.02	1.83	45.40
2013	17103.82	1357.38	53.01	91.83	2.04	44.20
2014	17239.07	1364.27	54.26	90.71	1.97	43.30
2015	17250.74	1371.22	55.50	90.66	2.15	41.10
2016	17441.79	1378.67	56.74	90.51	2.10	40.10
2017	18111.96	1386.40	57.96	89.68	2.08	40.50

oxidation rate of energy combustion; and 44 and 12 are the molecular weights of CO_2 and carbon, respectively.

According to IPCC guidelines for national greenhouse gas inventory [31], carbon dioxide emissions coefficients of various energies are shown in Table 4. Since electric power is a secondary energy and 70% of China's electric power is coal power, this paper converts energy consumption volumes of electric power into equivalent standard coal and, then, converts the carbon dioxide emissions of standard coal into those of electric power.

According to statistics data for energy consumptions of the Chinese transportation industry from the International Energy Agency, as well as the carbon dioxide emissions coefficients of various energies described in Table 4, calculated by means of formula (3), the carbon dioxide emissions volumes of Chinese transportation industry from 1973 to 2017 are, finally, obtained, as shown in Figure 1.

3.2. Prediction and Analysis of Influencing Factors. When using the SVR model to predict carbon dioxide emissions of the Chinese transportation industry, it is generally required to set the future value of the influencing factors reasonably to ensure the accuracy of prediction. However, the error of

Table 4: Carbon dioxide emissions coefficients of transportation and energy.

Types of energy	Average low calorific value (V_i) (kJ/toe)	Carbon oxidation rate (R_i) (%)	CO_2 emission factor (F_i) ($kgCO_2/GJ$)
Coal	20908	1	94.6
Oil products	43070	1	72.35
Biomass energy	42338	1	75.18
Natural gas	38931	1	56.1
Electric power	—	—	—

Note: data source: the Intergovernmental Panel on Climate Change (IPCC) 2006 edition.

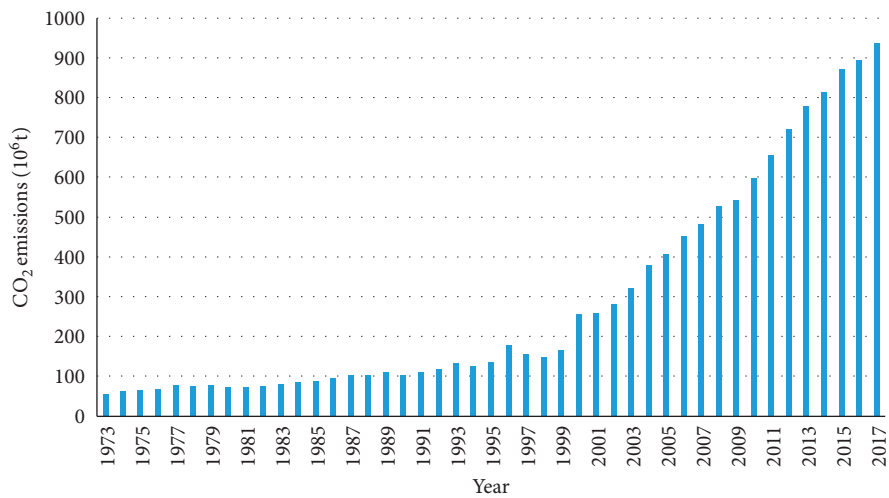


Figure 1: Carbon dioxide emissions from the Chinese transportation industry from 1973 to 2017.

setting the future value of each influencing factor has a little influence on the prediction of the final dependent variable.

3.2.1. Prediction of GDP. In 2015, the research group of the Economic Research Institute of the National Development and Reform Commission released the interim results of the *Research on Chinese Development Environment, Development Trend, and Strategic Thinking during the 13th Five-year Plan Period*, which concluded that the average growth rate of the Chinese economy was around 6.5% during the 13th five-year plan period. According to the *Economic Blue Book* released by the Chinese Academy of Social Sciences in 2018, the Chinese economy was expected to grow by about 6.6% in the year, continuing an overall stable and healthy development trend, and the Chinese GDP growth rate was forecast to be around 6.4% in 2019, 0.2 percentage points lower than that of the previous year. According to the *Chinese Economic Report* in 2018, the annual average growth rates of the Chinese economy were predicted to be 6.5% in 2016–2020 and drop to 5.0% in 2021–2035. In February 2019, the China Development Research Foundation predicted that the Chinese economic growth rate would be more than 6.0 percent by 2021 and would drop to around 5.0 percent after 2022. Based on the abovementioned predicted results, this paper sets the growth rates of Chinese economical development in 2018–2025, 2026–2030, 2031–2035, 2036–2040, 2041–2045, and 2046–2050 as 6.0%, 5.5%, 5.0%, 4.5%, 4.0%, and 3.5%, respectively. The setting value of the growth rate of Chinese GDP in each period of the high-carbon scenario is 0.3% higher than those of the baseline scenario, while the setting value of the growth rate of Chinese GDP in each period of the low-carbon scenario is 0.3% lower than those of the baseline scenario.

3.2.2. Population Prediction. According to the *report on national population development strategy research* released by the Chinese group of national population development strategy research, the total population of China will reach 1.45 billion by 2020, and it is predicted that the total population of China will reach a peak of about 1.5 billion around 2033 [32]. According to the *national population development plan* (2016–2030) issued by the state council in 2016, the annual natural growth rate during the 12th five-year plan period remained at about 5‰, and the total population growth in the following 15 years presented an inertia reduce, reaching a peak around 2030. *The world population outlook* 2019*: development summary* released by the United Nations in 2019 predicts the population trend of China. The population of China in 2018 was 1.395 billion people. The UN's medium fertility model predicts that China will reduce by about 30 million people by 2050, and the UN's low fertility model predicts that China will reduce by 136 million people by 2050. The Institute of Population and Labor Economics of the Chinese Academy of Social Sciences and the Social Sciences Academic Press jointly published *the green book on population and labor: Chinese population and labor issues No. 19*, which predicted that the Chinese population would reach a peak of 1.442 billion in 2029, enter a

continuous negative growth from 2030, and reduce to 1.364 billion in 2050 [33].

Based on the results predicted above, this paper sets up the change rates of the Chinese population in 2018–2025, 2026–2030, 2031–2035, 2036–2040, 2041–2045, and 2046–2050 as 0.4%, 0.2%, −0.2%, −0.3%, −0.4%, and −0.5%, respectively, under the set baseline scenarios. The setting value of the change rate of population in each period of the high-carbon scenario increases by 0.1% compared with those in the baseline scenario, while the setting value of the change rate of population in each period of the low-carbon scenario decreases by 0.1% compared with those in the baseline scenario.

3.2.3. Prediction of the Urbanization Rate. Changes of the Chinese urbanization rate over the years are shown in Figure 2. According to the 2013 *China Human Development Report* released by the United Nations Development Program (UNDP), the urbanization rate in China will reach 70% by 2030 [34]. According to the *National New Urbanization Plan* (2014–2020) issued by the Chinese State Council in 2014 and the *13th Five-year Plan for National Economic and Social Development* in 2016, the Chinese urbanization rate will reach 60% by 2020. In 2019, the Chinese Academy of Social Sciences Institute of Urban Development and Environment Research and the Social Sciences Academic Press jointly released the *Urban Blue Book: China City Development Report No. 12*, which points out that the Chinese urbanization rate reached 59.58% in 2018. It is about to enter in the late stage of urbanization. By 2050, the Chinese urbanization rate will reach 80% and the urbanization still has a relatively large development space and potential.

Based on the results predicted above, this paper sets the annual average growth rate of the urbanization rate from 2018 to 2025 as 1.5%, and the urbanization rate will reach 60.61% by 2020. The annual average growth rate from 2026 to 2030 will be 1.3%, and the urbanization rate will be 69.65% by 2030. The annual growth rates of 2031–2035, 2036–2040, 2041–2045, and 2046–2050 are 1.1%, 0.9%, 0.8%, and 0.7%, respectively. In the high-carbon scenario, the setting value of the annual average growth rate of urbanization in each period increases by 0.1% compared with those of the baseline scenario, while in the low-carbon scenario, the setting value of the annual average growth rate of urbanization in each period decreases by 0.1% compared with those of the baseline scenario.

3.2.4. Prediction of Energy Consumptions Structure. The proportion of fossil energy consumption in the Chinese transportation industry was 100% in 1973 and 89.68% in 2017, showing an overall downward trend. *The energy strategic action plan* (2014–2020), released by the state council in 2014, aims to increase the proportion of nonfossil energy in primary energy consumptions to 15% and that of natural gas to more than 10% by 2020. Similarly, the 13th five-year energy development plan, released by the national development and reform commission in 2016, pointed out that efforts should be made to promote the transformation of energy production and utilization, build a clean, low-carbon, safe, and efficient modern energy supply and demand system, and increase the proportion of nonfossil energy consumption to 15% by 2020. In 2015, the Chinese government issued the *Strengthening Action on Climate Change- Chinese Independent Contribution Rate*, which proposed that the proportion of nonfossil energy in primary energy consumption should reach about 20% by 2030. In 2018, the China Petroleum Institute of Economics and Technology released the *World and Chinese Energy Outlook* 2050, which indicated that the proportion of Chinese nonfossil energy will reach about 23% in 2030, and coal, oil, and nonfossil energy will account for one-third, respectively, by 2050.

Therefore, according to the abovementioned policy planning and the prediction of relevant institutions, this paper sets, under the baseline scenario, the change rates of energy consumptions structure in 2018–2025, 2026–2030, 2031–2035, 2036–2040, 2041–2045, and 2046–2050 are, respectively, −2.0%, −1.6%, −1.3%, −1.0%, −0.8%, and −0.6%. In the high-carbon scenario, the setting value of the change rate of energy consumption structure in each period increases by 0.5% compared with those of the baseline scenario, while the setting value of the change rate of energy consumption structure in each period of the low-carbon scenario decreases by 0.5% compared with those of the baseline scenario.

3.2.5. Prediction of Energy Intensity. The change trend and annual change rate of energy intensity for the Chinese transportation industry in each year within the research range are shown in Figure 3. From 1973 to 2017, the energy intensity of the Chinese transportation industry witnessed a fluctuant change, and the annual growth rate also fluctuated relatively large. From 1976 to 1990, it showed a relatively large decline; from 1991 to 2003, it changed with fluctuation; in 2003, the energy intensity reached the peak, and since then, the energy intensity fluctuated slowly and had a downward trend.

The Chinese *13th Five-year Plan for National Economic and Social Development* in 2016 called for a 15% reduction in energy intensity during the 13th five-year plan period. According to the existing policies and energy intensity trends, this paper sets that, under the baseline scenario, the change rates of energy intensity in 2018–2025, 2026–2030, 2031–2035, 2036–2040, 2041–2045, and 2046–2050 are −2.0%, −1.8%, −1.6%, −1.4%, −1.2%, and −1.0%, respectively. The setting value of the change rate of energy intensity in each period of the high-carbon scenario increases by 0.2% compared with those in the baseline scenario, while the setting value of the change rate of energy intensity in each period in the low-carbon scenario decreases by 0.2% compared with those in the baseline scenario.

3.2.6. Prediction of the Industrial Structure. The industrial structure in this paper refers to the proportion of the secondary industry in GDP. Figure 4 shows the trend of the

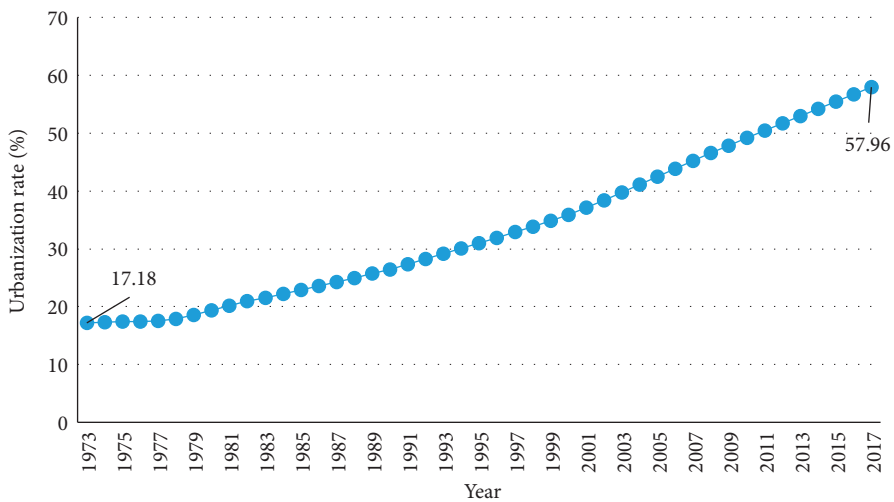


Figure 2: Chinese urbanization rates over the years.

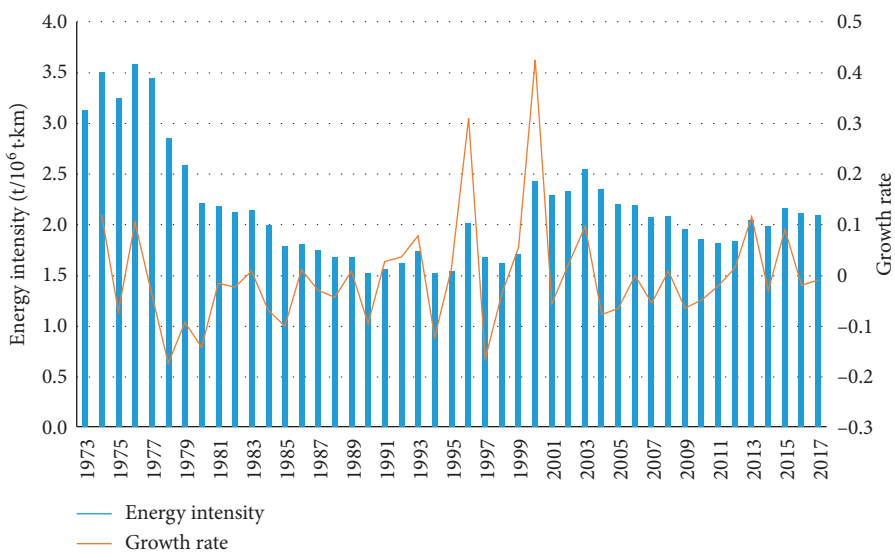


Figure 3: The energy intensity and its growth rate.

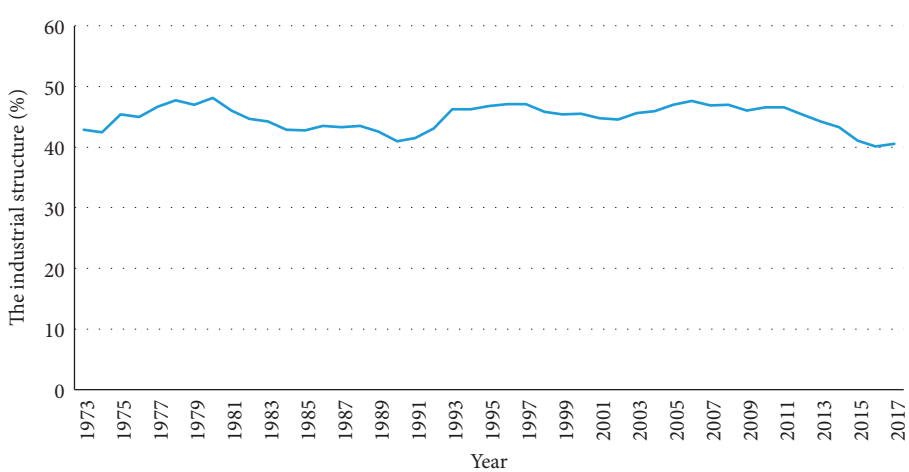


Figure 4: The proportion of secondary industry in Chinese GDP from 1973 to 2017.

proportion of the secondary industry in GDP in China from 1973 to 2017.

On the whole, the proportion of secondary industry changes relatively gradually, basically maintaining in the range of 40–50%, and the annual average rate of change from 1973 to 2017 is only −0.13%. After 2011, the decline trend was obvious, and the annual average rate of change from 2011 to 2017 was −2.28%. In 2017, the proportion of secondary industry was 40.5%. With the development of Chinese economy and the upgrading and transformation of the industrial structure, the proportion of the secondary industry in China will continue to decline in the future. In 2013, *the China* 2030 was released by the joint research group of the World Bank and the Development Research Center of the state council. It predicted that, in the absence of major changes in the international situation and no major impact of reform, the proportion of secondary industry in China will decline and reach 34.6% of GDP in 2030.

At present, the Chinese economy is in a new normal period of "speed change, structure optimization, and motivation transformation," and the trend of accelerating the transformation of the economy from industry domination to service industry domination is more obvious. Taking into account the historical change trend of the proportion of secondary industry in China and the prediction results of relevant institutions, the change rates of the industrial structure in 2018–2025, 2026–2030, 2031–2035, 2036–2040, 2041–2045, and 2046–2050 under the baseline scenario set in this paper are −1.5%, −1.4%, −1.3%, −1.2%, −1.1%, and −1.0%, respectively. In the high-carbon scenario, the setting value of the change rate of the industrial structure in each period increases by 0.2% compared with those of the baseline scenario, while in the low-carbon scenario, the setting value of the change rate of the industrial structure in each period decreases by 0.2% compared with those of the baseline scenario.

3.2.7. Summary of Growth Rates Setting of the Influencing Factors. The growth rate setting of each influencing factor under the three scenarios is shown in Table 5.

4. Discussion

4.1. Operation Results of the SVR Model. Firstly, the influence factors of GDP, population, and urbanization rate of carbon dioxide emissions in transportation industry are taken as the input data of the model and the volumes of carbon dioxide emissions as the output data, and then, the data of 37 years from 1973 to 2009 are used as the sample set for simulation and emulation.

Secondly, according to the steps of the nonlinear ε-SVR prediction model, the sample data of the 37 years are normalized and the prediction model is established. When conducting sample training and prediction, the penalty factor c and parameter g of kernel function need to be determined. MATLAB software is used to process relevant data. The initial value of ε is set at 0.01, and the value of c and g both range in $[2^{-8}, 2^{8}]$. After repeated tests, when c is set at 0.5743 and g at 1.0353, the predicted data of sample set are fitted and regressed with the actual data, and the mean square error (MSE) is only 0.000454, indicating that the predicted results are relatively satisfactory. Figure 5 is the effect diagram for parameter selection of grid algorithm, and Figure 6 shows the comparison between the training sample and the actual value.

Finally, in order to verify the validity of the established model, the data from 2010 to 2017 are taken as test samples for prediction, the predicted data and actual data are fitted for regression, and the mean square error is 0.088836. All these indicate that the predicted results are close to the true value of carbon dioxide emissions, that is, the SVR model has an excellent prediction effect on carbon dioxide emissions, and therefore, it can be used as an effective method to predict the carbon dioxide emissions of the Chinese future transportation industry.

4.2. Predicted Results on the Peak Values of Carbon Dioxide Emissions. Three scenarios were predicted by using the established SVR model, and the obtained results are shown in Figure 7 and Table 6. It can be found from the predicted results that, in the baseline scenario, the carbon dioxide emissions of the Chinese transportation industry are still in a growing trend from 2018 to 2040, with a peak of 1365.71 million tons in 2040. The peak value corresponding to the low-carbon scenario is 1115.43 million tons, appearing in 2035. In the high-carbon scenario, the total carbon dioxide emissions are on the rise and peak in 2048, but the carbon dioxide emissions are larger than those of other scenarios, at 1738.18 million tons.

Currently, a few scholars have predicted the peak values of carbon dioxide emissions in the Chinese transportation industry. Chen et al. used the Carbon Kuznets Curve (CKC) as the theoretical model to predict the peak values, and the results showed that Chinese carbon dioxide emissions will reach the peak in 2036. Among them, the peak time of the industrial sector is 2031, that of the construction sector is 2035, that of the transportation sector is 2043, and that of agriculture sector is 2026 [35]. Thus, the peak time of the baseline scenario predicted in this paper is close to the prediction result of Chen et al. In addition, Chinese officials at the world climate conference in Copenhagen predicted that Chinese greenhouse gas emissions will peak between 2030 and 2040. Yuan et al., taking into account the changing trends of Chinese future population, GDP, industrial structure, urbanization, energy intensity and energy consumption, predicted that Chinese carbon dioxide emissions will reach a peak of 9.2 to 9.4 billion t from 2030 to 2035 [36]. Generally speaking, the arrival time of the peak of carbon dioxide emissions in the Chinese transportation industry lags behind the national total peak time, as well as those of the agricultural and industrial sectors. By analyzing the predicted results of the abovementioned literatures, it can be inferred that the predicted results of this paper are reasonable to some extent.

4.3. Sensitivity Analysis. To analyze the effect of individual factor influencing on the carbon dioxide emissions of the Chinese transportation industry, this thesis, based on the

Table 5: Growth rates setting of carbon dioxide emissions' influencing factors in the Chinese transportation industry.

Scenario	Year	Growth rate setting (%)					
		GDP	Population	Urbanization rate	Energy consumptions structure	Energy intensity	Industrial structure
High-carbon scenario	2018–2025	6.3	0.5	1.4	−1.5	−1.8	−1.3
	2025–2030	5.8	0.3	1.2	−1.1	−1.6	−1.2
	2031–2035	5.3	−0.1	1.0	−0.8	−1.4	−1.1
	2036–2040	4.8	−0.2	0.8	−0.5	−1.2	−1.0
	2041–2045	4.3	−0.3	0.7	−0.3	−1.0	−0.9
	2046–2050	3.8	−0.4	0.6	−0.1	−0.8	−0.8
Baseline scenario	2018–2025	6.0	0.4	1.5	−2.0	−2.0	−1.5
	2025–2030	5.5	0.2	1.3	−1.6	−1.8	−1.4
	2031–2035	5.0	−0.2	1.1	−1.3	−1.6	−1.3
	2036–2040	4.5	−0.3	0.9	−1.0	−1.4	−1.2
	2041–2045	4.0	−0.4	0.8	−0.8	−1.2	−1.1
	2046–2050	3.5	−0.5	0.7	−0.6	−1.0	−1.0
Low-carbon scenario	2018–2025	5.7	0.3	1.6	−2.5	−2.2	−1.7
	2025–2030	5.2	0.1	1.4	−2.1	−2.0	−1.6
	2031–2035	4.7	−0.3	1.2	−1.8	−1.8	−1.5
	2036–2040	4.2	−0.4	1.0	−1.5	−1.6	−1.4
	2041–2045	3.7	−0.5	0.9	−1.3	−1.4	−1.3
	2046–2050	3.2	−0.6	0.8	−1.1	−1.2	−1.2

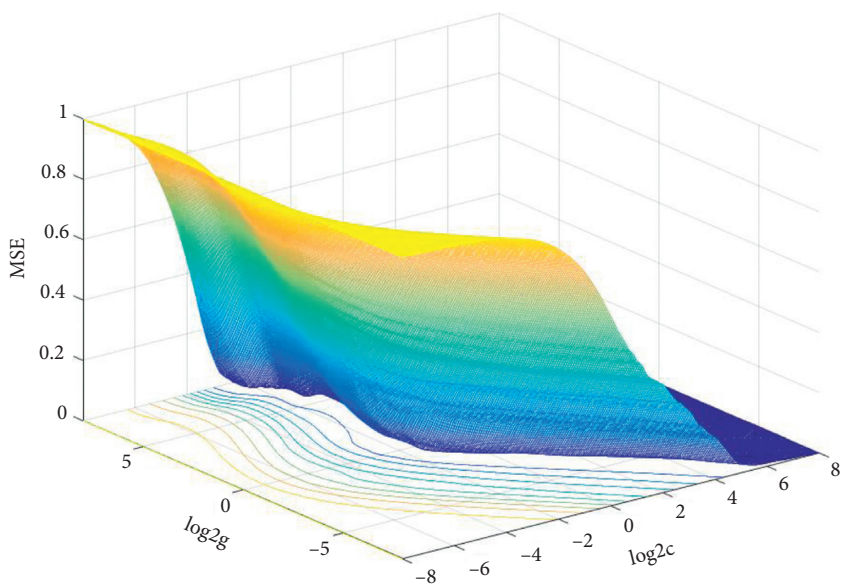


Figure 5: The effect diagram for parameter selection of grid algorithm (3D).

baseline scenario, changes a kind of factors affecting the change rate in sequence, namely, on the basis of the baseline rate changes −10% and 10%, respectively. When the change rates of other factors are the values of rate settings of the baseline scenario, it quantitatively analyzes of the factors that affect the carbon dioxide emissions of the Chinese transportation industry. The results are shown in Table 7.

Overall, GDP, population, and other factors have a certain influence on carbon dioxide emissions. Among all the influencing factors, the change of energy consumption structure has the greatest influence on carbon dioxide emissions. With the rate of energy consumption structure decreasing by 10%, the peak value of carbon dioxide emissions decreases by 4.13% compared with that of the baseline scenario, and the total carbon dioxide emissions will decrease by 3.39% from 2018 to 2050. Population is next. With the change rates of population falling by 10%, the peak value of carbon dioxide emissions falls by 4.07%, and the total carbon dioxide emissions will fall by 3.32% between 2018 and 2050. Among them, GDP and the urbanization rate have less influence on carbon dioxide emissions. The change rate of GDP reduces by 10%, and the total carbon dioxide emissions will reduce by 2.37% from 2018 to 2050. The change rate of urbanization rate reduces

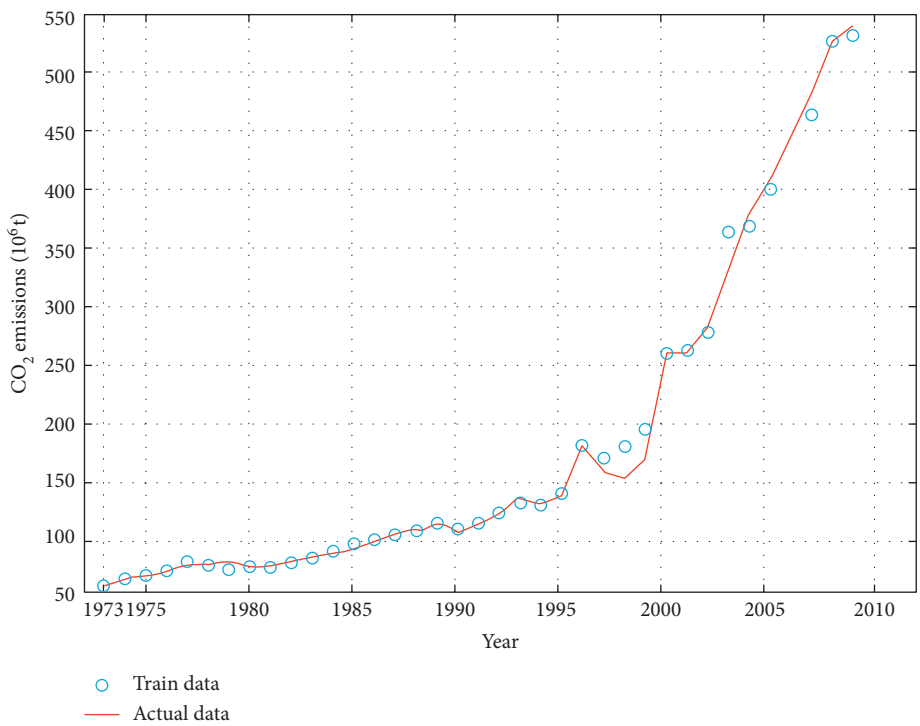


Figure 6: Comparison of the predicted value of the training sample with the actual value.

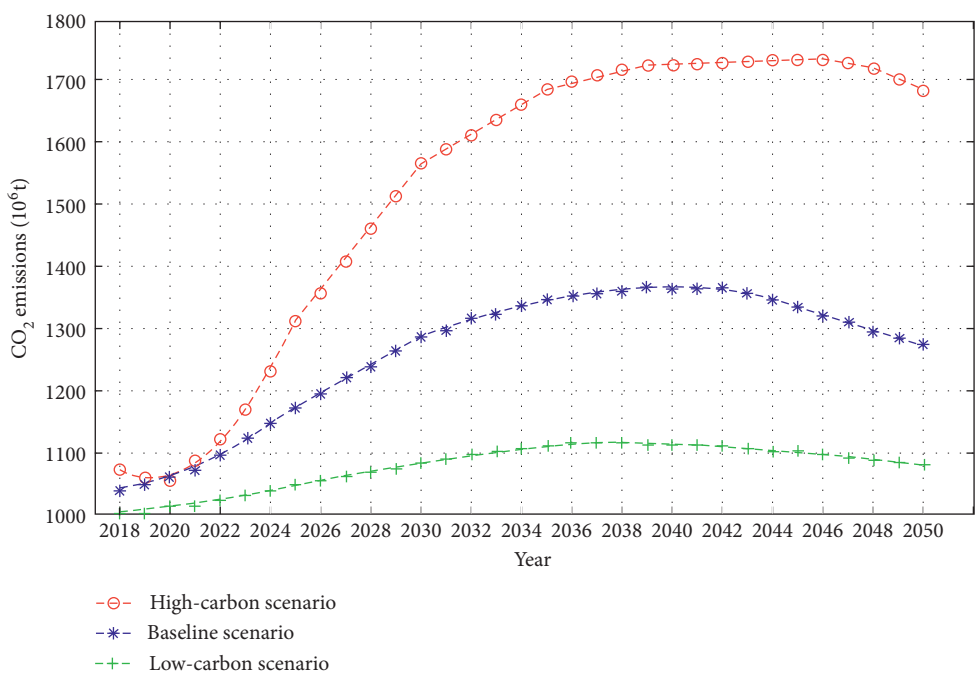


Figure 7: The trend of prediction results of carbon dioxide emissions under three scenarios.

Table 6: Prediction on peak values of different scenarios.

Scenario	Peaking time (year)	Peak value (10^6 t)
High-carbon scenario	2046	1738.18
Baseline scenario	2040	1365.71
Low-carbon scenario	2036	1115.43

Table 7: Carbon dioxide emissions of each factor at different change rates.

Factor	Change rate (%)	Peak value (10^6 t)	Change of the peak value (%)	Total carbon dioxide emissions from 2018 to 2050 (10^6 t)	Change of total volume (%)
GDP	−10	1358.13	−0.55	40610.82	−2.37
	10	1420.17	3.99	42988.74	3.34
Population	−10	1310.10	−4.07	40216.32	−3.32
	10	1424.80	4.33	43066.51	3.53
Urbanization rate	−10	1311.01	−4.00	40240.50	−3.26
	10	1423.63	4.24	43034.93	3.45
Energy consumptions structure	−10	1309.34	−4.13	40189.08	−3.39
	10	1425.78	4.40	43096.09	3.60
Energy intensity	−10	1310.57	−4.04	40226.69	−3.30
	10	1424.21	4.28	43052.69	3.50
Industrial structure	−10	1310.88	−4.01	40228.06	−3.31
	10	1423.81	4.25	43051.13	3.49

by 10%, and the total carbon dioxide emissions from 2018 to 2050 will reduce by 3.26%.

From Table 7 and the abovementioned analysis, it can be seen that, in order to reduce the carbon dioxide emissions of the transportation industry, it is necessary to reasonably control all of the influencing factors. First, we reduce the energy consumption structure and energy intensity of the transportation industry, improve the efficiency of clean energy, and actively promote the use of new energy and clean energy vehicles and ships. Second, we change the pattern of economic growth, appropriately reduce the speed of economic development, and strive to achieve coordinated development between economic growth and environmental protection. At the same time, we need to optimize the industrial structure, move the industry toward the middle and high end, achieve high-quality development, and reduce the demand for transportation, thereby reducing the carbon dioxide emissions. Although the change of urbanization rate has a relatively small influence on the carbon dioxide emissions of the transportation industry, urbanization has changed people's lifestyle and their demands for energy are increasing. China is in the stage of urbanization, with the population moving from rural areas to cities and towns, and the change of people's production and lifestyle affects the change of carbon dioxide emissions. Therefore, the rational development of urbanization also has an important influence on the reduction of carbon dioxide emissions.

5. Conclusions and Suggestions

5.1. Conclusions. Taking the Chinese transportation industry as the research object, this paper selects six major factors that affect the carbon dioxide emissions of the Chinese transportation industry, GDP, population, urbanization rate, energy consumption structure, energy intensity, and industrial structure, and establishes the prediction model for the peak of carbon dioxide emissions based on SVR. The mean square error of the prediction model is 0.000454, indicating a relatively high degree of coincidence of the model. The predicted results show that, under the low-carbon scenario, the carbon dioxide emissions of the Chinese transportation industry will peak at 1115.43 million tons in 2036. Under the baseline scenario, it will reach a peak of 1365.71 million tons in 2040. In the high-carbon scenario, the peak will occur in 2046 and the carbon dioxide emissions will be 1738.18 million tons. Finally, the influence of a single factor on the carbon dioxide emissions of the Chinese transportation industry is analyzed, which indicates that the change of each factor will have a certain influence on the peak of carbon dioxide emissions and the total carbon dioxide emissions from 2018 to 2050.

5.2. Suggestions. Since the peak time and total carbon dioxide emissions of the Chinese transportation industry vary greatly under different scenarios, major factors affecting the growth of carbon dioxide emissions must be controlled in order to promote the early peak time of the Chinese transportation industry. First, the pattern of economic growth is changed and the speed of economic development is appropriately reduced. The results of this paper show that the slowdown of economic growth is one of the main factors contributing to reducing the peak and total carbon dioxide emissions of the Chinese transportation industry. Therefore, China should gradually change the mode of economic growth, appropriately reduce the speed of economic development, and strive to achieve the coordinated developments of economic growth and environmental protection. Second, the energy intensity of the transportation industry is reduced and the utilization rate of clean energy is improved. We speed up the optimization of the structure of the transportation industry, reduce the volumes of bulk goods transported by road, increase the volumes of bulk goods transported by rail and waterways, substantially increase the volumes of multiple modes of combined transportation by port, railway, and container transportation, and reduce energy intensity. We actively promote the use of new and clean energy vehicles and ships and control CO_2 emissions from the transportation industry. Third, we improve the industrial structure, actively promote the optimization and upgrading of the industrial structure, develop strategic emerging industries and modern service industries, and move the industry to the medium-high end and achieve high-quality development, so as to reduce the demand for transportation and reduce carbon

dioxide emissions. Fourth, we reasonably control the traffic carbon dioxide emissions caused by the increase of the urbanization rate. We encourage residents to travel in a green way, gradually build a green travel structure with public transportation as the main part and walking and cycling as the auxiliary part, and reduce the frequency of car use.

Data Availability

In this paper, population and urbanization rate come from World Development Indicators (WDI), and the energy consumption structure comes from the International Energy Agency (IEA). Here is the percentage of fossil energy consumption in transportation industry accounting for the total energy consumptions. Total energy consumption in the Chinese transportation sector comes from the website of IEA. The rest of the data come from the comprehensive and publicly published Chinese Statistical Yearbook of past years. The sample interval is from 1973 to 2017.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Changzheng Zhu and Meng Wang conceived the study, wrote original draft, and contributed to all aspects of this work. Wenbo Du analyzed the data and gave some useful suggestions to this work.

Acknowledgments

The authors thank Dawei Gao for his help in participation in data collection and Lijiao Qin for her help in the English editing. This research was funded by the National Social Science Foundation in China (Grant no. 19BJY175), Shaanxi Natural Science Foundation project (Grant no. 2019JQ-533), Xi'an Science and Technology Plan (Grant no. 2019111913RKX003SF007-10), and Innovation fund for graduate students of Xi' an University of Posts and Telecommunications (Grant no. CXJJWY2018096).

References

[1] L. Shipper, L. Scholl, and L. Price, "Energy use and carbon emissions from freight in 10 industrialized countries: an analysis of trends from 1973 to 1992," *Transportation Research Part D: Transport and Environment*, vol. 2, pp. 57–76, 1997.

[2] G. R. Timilsina and A. Shrestha, "Transport sector CO_2 emissions growth in Asia: underlying factors and policy options," *Energy Policy*, vol. 37, no. 11, pp. 4523–4539, 2009.

[3] G. R. Timilsina and A. Shrestha, "Factors affecting transport sector CO_2 emissions growth in Latin American and Caribbean countries: an LMDI decomposition analysis," *International Journal of Energy Research*, vol. 33, no. 4, pp. 396–414, 2009.

[4] B. Talbi, "CO_2 emissions reduction in road transport sector in Tunisia," *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 232–238, 2017.

[5] C. Zhu and D. Gao, "A research on the factors influencing carbon emission of transportation industry in "the belt and road initiative" countries based on Panel data," *Energies*, vol. 12, no. 12, p. 2405, 2019.

[6] Y. Liang, D. Niu, H. Wang, and Y. Li, "Factors affecting transportation sector CO_2 emissions growth in China: an LMDI decomposition analysis," *Sustainability*, vol. 9, no. 10, p. 1730, 2017.

[7] Y. Wang, Y. Zhou, L. Zhu, F. Zhang, and Y. Zhang, "Influencing factors and decoupling elasticity of China's transportation carbon emissions," *Energies*, vol. 11, no. 5, p. 1157, 2018.

[8] Q. Du, Q. Su, Q. Yang, X. Y. Feng, and J. Yang, "Path analysis method of driving factors of carbon emissions for Chinese transportation industry," *Journal of Traffic and Transportation Engineering*, vol. 17, pp. 143–150, 2017.

[9] Y. Li, "Analysis of the influencing factors of carbon emission from Anhui Transportation Industry based on LMDI," *Environmental Protection*, vol. 5, pp. 5–8, 2019.

[10] R. J. Zhang, H. Z. Dong, Y. G. Han, and X. Li, "The spatial correlation analysis of influence factors on carbon emissions from the energy consumption," *Journal of Shandong University of Science and Technology*, vol. 34, pp. 33–39, 2020.

[11] A. Zhou and K. Y. Wu, "Driving factors analysis of traffic carbon emissions in Shanghai," *Journal of Hefei University of Technology (Natural Science)*, vol. 43, pp. 264–269, 2020.

[12] H. Li, Y. Lu, J. Zhang, and T. Wang, "Trends in road freight transportation carbon dioxide emissions and policies in China," *Energy Policy*, vol. 57, pp. 99–106, 2013.

[13] B. Xu and B. Lin, "Carbon dioxide emissions reduction in China's transport sector: a dynamic VAR (vector autoregression) approach," *Energy*, vol. 83, pp. 486–495, 2015.

[14] L. Wu, S. Kaneko, and S. Matsuoka, "Driving forces behind the stagnancy of China's energy-related CO_2 emissions from 1996 to 1999: the relative importance of structural change, intensity change and scale change," *Energy Policy*, vol. 33, no. 3, pp. 319–335, 2005.

[15] Y. E. Zhu, L. F. Li, S. S. He, H. Li, and Y. Wang, "Peak year prediction of Shanxi Province's carbon emissions based on IPAT modeling and scenario analysis," *Resources Science*, vol. 38, pp. 2316–2325, 2016.

[16] S. N. Qu and C. X. Guo, "Forecast of China's carbon emissions based on STIRPAT model," *China Population Resources and Environment*, vol. 20, pp. 10–15, 2010.

[17] Y. Z. Zhu, "Analyses on energy development and carbon exhaustion according to circumstances in future China's communications and transportation," *China Industrial Economics*, vol. 12, pp. 30–35, 2001.

[18] Q. Yang, R. H. Zhu, and X. Q. Zhao, "Calculation decoupling analysis and scenario prediction of carbon emissions of transportation in China," *Journal of Chang'an University*, vol. 34, pp. 77–83, 2014.

[19] J. Z. Zhang, X. C. Wang, Q. L. Tai, R. F. Xie, and Z. B. Chen, "Forecasting of energy demands and carbon emission of transportation in Hainan province," *Natural Science Journal of Hainan University*, vol. 35, pp. 164–170, 2017.

[20] P. R. Ehrlich and J. P. Holdren, "Critique," *Bulletin of the Atomic Scientists*, vol. 28, no. 5, pp. 16–27, 1972.

[21] T. Dietz and E. A. Rosa, "Rethinking the environmental impacts of population, affluence and Technology," *Human Ecology Review*, vol. 1, pp. 277–300, 1994.

[22] A. U. Gazi, A. Khorshed, and G. Jeff, "Ecological footprint and regional sustainability: a review of methodologies and results," in *Proceedings of the 37th Annual Conference of the*

Australia and New Zealand Regional Science Association International (ANZRSAI 2013), Hervey Bay, Australia, December 2013.

[23] S. F. Lin, S. Y. Wang, D. Marinova, and D. T. Zhao, "Improvement and application of STIRPAT model," *Statistics & Decisions*, vol. 34, pp. 32–34, 2018.

[24] Z. H. Chen, X. Q. Ye, and P. Huang, "Estimating carbon dioxide (CO_2) emissions from reservoirs using artificial neural networks," *Water*, vol. 10, p. 26, 2018.

[25] Y. F. Lin, H. M. Deng, and X. Y. Shi, "Application of BP neural network based on newly improved particle swarm optimization algorithm in in fitting nonlinear function," *Computer Science*, vol. S2, pp. 51–54, 2017.

[26] J. Cai and X. Ma, "Carbon emission prediction model of agroforestry ecosystem based on support vector regression machine," *Applied Ecology and Environmental Research*, vol. 3, pp. 6397–6413, 2019.

[27] L. Chen, J. H. Wang, T. He, Z. H. Zhou, Q. R. Li, and W. W. Yang, "Forecast study of regional transportation carbon emissions based on SVR," *Journal of Transportation Systems Engineering and Information Technology*, vol. 18, pp. 13–19, 2018.

[28] J. K. Song, "China's carbon emissions prediction model based on support vector regression," *Journal of China University of Petroleum*, vol. 36, pp. 182–187, 2012.

[29] L. M. Xue, X. Z. Zhang, B. K. Liu, and Y. G. Hu, "SVR-based prediction of carbon emissions from energy consumption in Hebei Province," *Coal Engineering*, vol. 49, pp. 165–168, 2017.

[30] B. R. Li, *Economic Forecast Theory, Method and Application*, Economy & Management Publishing House, Beijing, China, 2003.

[31] Intergovernmental Panel on Climate Change (IPCC), *IPCC Guidelines for National Greenhouse Gas Inventories 2006 Volume 2 Energy; Intergovernmental Panel on Climate Change*, Kanagawa, Japan, 2007.

[32] Z. X. Feng and A. J. Wang, "Comparative study of China regional carbon peak—based on national data and Shaanxi province," *Journal of Xi'an Jiaotong University*, vol. 36, pp. 96–104, 2016.

[33] C. W. Zhang, *Green Paper on Population and Labor: China's Population and Labor Problem Report No. 19*, Social Sciences Academic Press, Beijing, China, 1st edition, 2018.

[34] The United Nations Development Programme (UNDP), *China National Human Development Report 2013, Sustainable and Liveable Cities*, Toward Ecological Civilization, Beijing, China, 2013.

[35] X. Chen, C. Y. Shuai, Y. Wu, and Y. Zhang, "Analysis on the carbon emission peaks of China's industrial, building, transport, and agricultural sectors," *Science of the Total Environment*, vol. 709, 2020.

[36] J. Yuan, Y. Xu, Z. Hu, C. Zhao, M. Xiong, and J. Guo, "Peak energy consumption and CO_2 emissions in China," *Energy Policy*, vol. 68, pp. 508–523, 2014.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8866876, 10 pages
https://doi.org/10.1155/2020/8866876

Research Article

Identifying Big Five Personality Traits through Controller Area Network Bus Data

Yameng Wang,[1,2] **Nan Zhao**,[1] **Xiaoqian Liu**,[1] **Sinan Karaburun**,[3] **Mario Chen**,[4] **and Tingshao Zhu**[1]

[1]*CAS Key Laboratory of Behavioral Science, Institute of Psychology, Chinese Academy of Sciences, Beijing, China*
[2]*School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China*
[3]*BMW China Automotive Trading Ltd., Beijing, China*
[4]*BMW China Services Ltd., Beijing, China*

Correspondence should be addressed to Tingshao Zhu; tszhu@psych.ac.cn

Received 18 August 2020; Revised 30 September 2020; Accepted 6 October 2020; Published 20 October 2020

Academic Editor: Petr Dolezel

As adapting vehicles to drivers' preferences has become an important focus point in the automotive sector, a more convenient, objective, real-time method for identifying drivers' personality traits is increasingly important. Only recently has increased availability of driving signals obtained via controller area network (CAN) bus provided new perspectives for investigating personality differences. This study proposes a new methodology for identifying drivers' Big Five personality traits through driving signals, specifically accelerator pedal angle, frontal acceleration, steering wheel angle, lateral acceleration, and speed. Data were collected from 92 participants who were asked to drive a car along a pre-defined 15 km route. Using statistical methods and the discrete Fourier transform, some time-frequency features related to driving were extracted to establish models for identifying participants' Big Five personality traits. For these five personality trait dimensions, the coefficients of determination of effective predictive models were between 0.19 and 0.74, the root mean squared errors were between 2.47 and 4.23, and the correlations between predicted scores and self-reported questionnaire scores were considered medium to strong (0.56–0.88). The results showed that personality traits can be revealed through driving signals, and time-frequency features extracted from driving signals are effective in characterizing and identifying Big Five personality traits. This approach could be of potential value in the development of in-car integration or driver assistance systems and indicates a possible direction for further research on convenient psychometric methods.

1. Introduction

It has been shown that personality traits can be used to explore individuals' potential needs in different contexts, such as driving. Several studies have demonstrated that risky driving behaviors are positively associated with neuroticism and extraversion [1–3], but negatively associated with agreeableness, openness, and conscientiousness [1, 4, 5]. Furthermore, Shen et al. [6] found that positive driving behaviors are negatively correlated with neuroticism and positively correlated with openness, conscientiousness, extraversion, and agreeableness. Currently, there is a need for individualization of vehicles in the automotive industry with the aim of improving driving experiences [7]. Thus, it has become an important focal point [8] to adapt the vehicle to drivers' preferences (e.g., personality).

In the traditional method of measuring personality traits, self-report questionnaires, such as the 44-item Big Five Inventory (BFI-44), are used [9]. Although personality traits are relatively stable individual psychological variables that do not need to be measured frequently over a short period of time [10], relying on self-report questionnaires limits its potential to improve driving experiences in some scenarios. For instance, for nonfixed drivers (e.g., taxis, rental cars, and family cars) filling out a questionnaire before every time they drive not only does not meet a driver's need for vehicle

adaptation but also takes considerable time and concentration, which limits the availability and effectiveness of self-reported personality traits. Therefore, a more convenient, objective, real-time method to identify drivers' personality traits has become increasingly important.

Sensors and electronic control units (ECUs) have only recently become increasingly common in the automotive industry, because they not only guarantee optimal engine function, but also provide a large amount of almost real-time data about the car, driver, and surrounding environment. Various tools, including sensors, driving simulators, and controller area network (CAN) bus data logger, have been applied to conduct studies and many meaningful conclusions have been achieved. For instance, regarding the results of vehicle acceleration and steering behaviour analysis as indicators of driving safety, Wu et al. [11] attempted to determine the optimum design of pavement marking to reduce the rutting on asphalt pavements. Besides, for safety consideration, driving simulators have been used in studies where field operating tests cannot be carried out, such as the study investigating the safety of trucks under crosswind of tunnel and bridge sections [12]. Moreover, the CAN data have been used for the communication among ECUs mounted to a car [13], the tuning problem of digital proportional-integral-derivative parameters for a DC motor [14], and integrated motor-transmission powertrain systems [15]. Additionally, as one of the five protocols used in OBD-II vehicle standards, CAN technology has become the standard for automotive embedded systems [16]. The increased availability of rich driving data has provided new perspectives for investigating individual behaviors and psychological indicators.

With the advantage of high quality and fine data granularity of driving signals provided by in-vehicle sensors, many studies have been conducted based on these data. It has been demonstrated that driving signals can be used to recognize drunk driving behaviors [17], identify drivers [18], and detect anomalous driving [19]. Additionally, the capability of recording real-time driving information is soon used in other applications with the help of machine learning technology [20]. Furthermore, Wan et al. [21] attempted to detect anger states while driving based on multiple sensor signals using a least square support vector machine model (82.20% accuracy rate).

In summary, although there has been evidence that driving signals can reveal personality traits, the method of identifying personality traits based on driving signals has not been established in previous studies. It motivates our efforts to intensively explore the possibility of a solution for real-time identification of personality traits through driving signals. In this work, we aimed to construct feature sets from raw driving signals provided by in-vehicle sensors using CAN bus and identify Big Five personality traits based on these features using a machine learning approach.

2. Materials and Methods

In this section, a methodology with the aim of identifying personality traits through CAN bus data is proposed. Using statistical methods and the discrete Fourier transform, the features related to personality traits are extracted from raw driving signals provided by in-vehicle sensors using CAN bus in the time and frequency domains, respectively. These features will be then used to identify Big Five personality traits automatically by the linear regression, support vector regression, etc. In the study, a four-step procedure was conducted: (1) Data collection, (2) Data preprocessing, (3) Feature extraction and selection, and (4) Model training, as shown in Figure 1.

2.1. Data Collection

2.1.1. Experimental Settings. A BMW i3 test vehicle was equipped with a data logger to record the signals on the CAN bus at the sampling frequency of 10 data points per second for this study. We collected data from 92 participants (52 males and 40 females) who were recruited using convenience sampling from BMW China. All of the participants were asked to drive the BMW i3 test vehicle on a pre-defined route as shown in Figure 2. The pre-defined car route was 15 km and included traffic lights stop signs, surface streets etc. With this user consistent driving task, we wanted to eliminate interference information, so as to explore deeper insights between driving behavior and personality traits. To facilitate data analysis, we divided the route into different sub-routes according to road conditions, and an instructor sitting in the copilot recorded the time that the car passed through different sub-routes during the experiment.

Once the procedure of driving signals collection was done, each participant was required to complete the BFI-44 to measure their Big Five personality traits. The questionnaire consists 44 items and five subscales: openness (10 items); conscientiousness (9 items); extraversion (8 items); agreeableness (9 items); and neuroticism (8 items). Each item of BFI-44 is assessed on a Likert 5-point scale, ranging from 1 ("disagree strongly") to 5 ("agree strongly"). In this study, the Chinese version of the questionnaire was implemented. Its validity and reliability has been proved [22].

2.1.2. Signals Selection. Among the signals transmitted on the CAN bus, the analyses of this study focused on five signals recorded at the sampling frequency of 10 data points per second: accelerator pedal angle, frontal acceleration, steering wheel angle, lateral acceleration, and speed. Compared with other signals, these signals are not only more stable and easy to obtain on different types or models of vehicles, but also can reflect drivers' driving behavior from different aspects. For instance, accelerator pedal and steering wheel signals are the direct output of drivers that directly reflect the interaction between the driver and the vehicle [23]; speed and accelerations are measures of drivers' driving style [24] that can reflect drivers' specific driving preferences and habits, e.g., harsh accelerations or speeding. An example of these signals is shown in Figure 3.

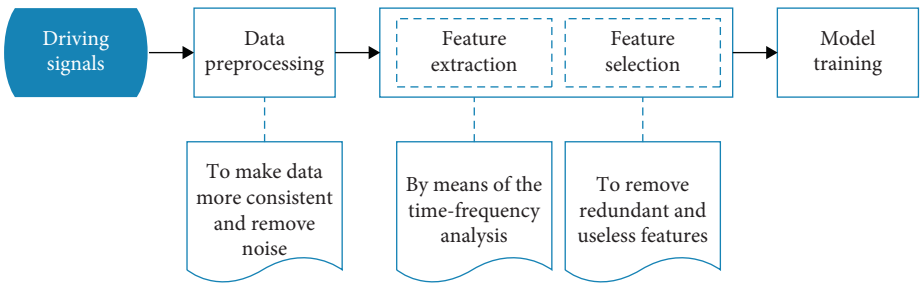


Figure 1: The procedure for identifying Big Five personality traits from driving signals.

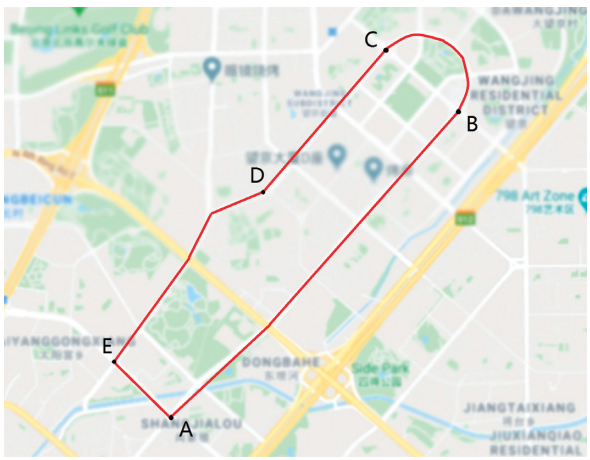


Figure 2: Pre-defined route during data collection (image blurred for anonymity purposes).

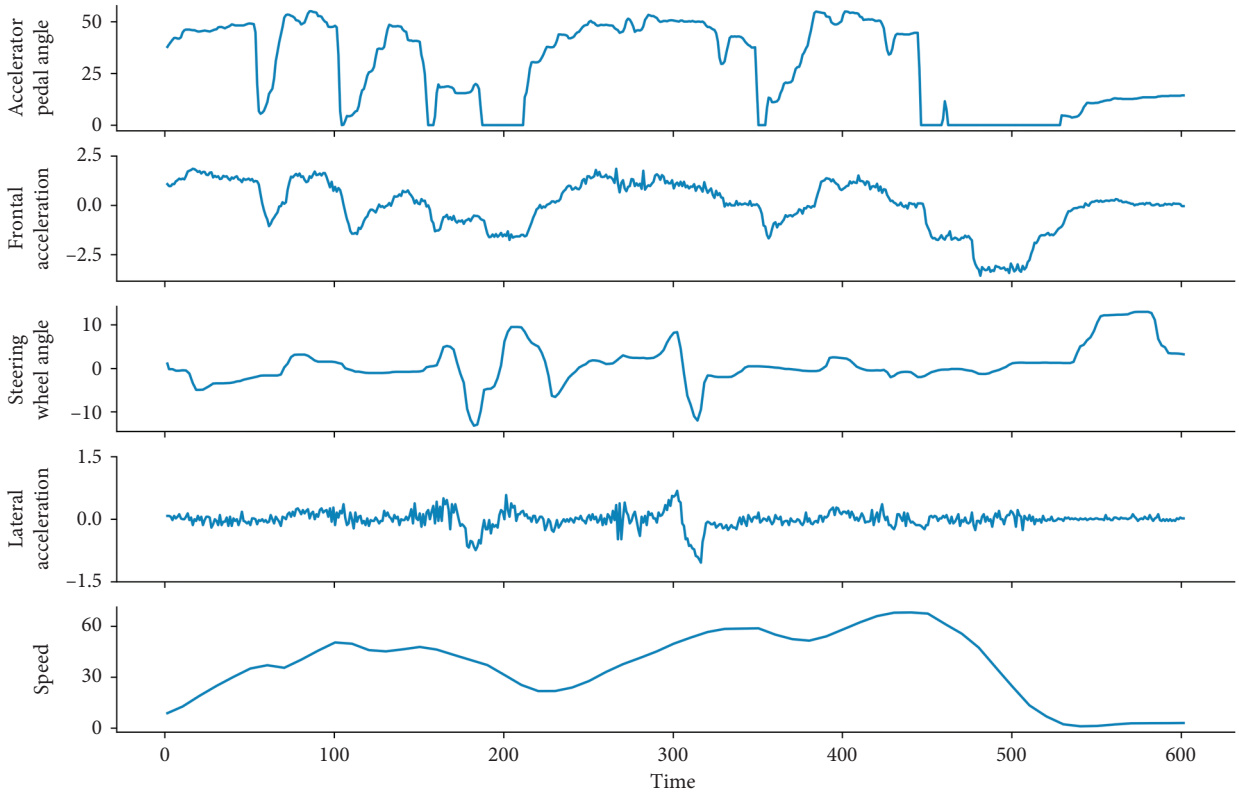


Figure 3: Example of signals acquired from the CAN bus.

2.2. Data Preprocessing. Raw driving signals with noisy and redundant information may bring more redundancy and complexity for models training and affect the performance of recognition models. Therefore, we need to preprocess the raw driving signals, which includes two steps: (1) data segmentation and (2) low-pass filtering.

2.2.1. Data Segmentation. Since driving under the same road conditions can be regarded as repetitive behaviors, large amounts of repetitive data may lead to low computational efficiency and data redundancy. In addition, it is difficult to guarantee the consistency of road conditions such as corners or curved roads in the actual driving environment. Then recognition models trained based on data obtained under such road conditions may have a poor generalization ability in practice. In this work, we analyzed driving signals of a straight sub-route from point A to point B (as shown in Figure 1). On average, participants took 26.03 minutes (SD = 7.48) to complete the course. For the consistency of driving data, we used driving signals for the first 9600 data points (16 minutes).

2.2.2. Low-Pass Filtering. As unexpected jolts or vibrations might cause some noise or high-frequency components in data collection, we should do the job of filtering on the raw driving signals as the signal processing. Gaussian filter is a low-pass filter, attenuating noises and high-frequency components in signal data [25]. We computed the convolution of each driving signal and the Gaussian filter, whose window length is 5, and whose coefficients are $g = (1/16)[1, 4, 6, 4, 1]$. The procedure of filtering is defined as

$$y(n) = \sum_{t=-\infty}^{\infty} x(t)g(n-t) = x(n) * g(n), \tag{1}$$

where x is the driving signal, $*$ stands for convolution, and g denotes the Gaussian filter. We take a fragment of the frontal acceleration as an example. After low-pass filtering, the filtered data (See Figure 4(b)) are smoother compared to the raw data (See Figure 4(a)). And many little fluctuations and burrs shown in the red circle in Figure 4(a) are removed.

2.3. Feature Extraction and Selection. After data preprocessing, we then need to extract and select features from driving signals that can effectively characterize the Big Five personality traits. Specifically, using the time-frequency analysis method, we first extracted features in the time and frequency domains, respectively. And then we find and remove redundant information from these features by dimensionality reduction and feature selection.

2.3.1. Temporal Domain Features Extraction. Temporal domain information related to the statistical value of driving signals (e.g., mean value, median value, and standard deviation value) was used to characterize drivers' behavior patterns. Since the global statistical value of signals cannot reflect the details of driving behavior, this information was integrated into a given sliding temporal window. Specifically, in a temporal window of width w, we defined the set of data $U_{j\in I_i} x_j$, $I_i = \{i+1, i+2, \cdots, i+w\}$ and the following features:

(1) Moving median: the median value of the set.

(2) Moving mean: the mean value of the set

(3) Moving standard deviation: the standard deviation value of the set.

To exam linear dependence of a signal, we estimated autocorrelation and partial autocorrelation of different lags. Specifically, autocorrelation is the correlation of a signal with a delayed copy of itself [26], which is defined as

$$r_k = \frac{\sum_{t=1}^{n-k} (x_t - \mu)(x_{t+k} - \mu)}{\sum_{t=1}^{N} (x_t - \mu)^2}, \tag{2}$$

where n refers to the length of the signal, μ refers to mean of the signal, and k refers to the lag. Partial autocorrelation gives the partial correlation of a stationary time series with its own lagged values [26], which is defined as

$$p_k = \frac{\operatorname{cov}(x_t, x_{t-k} | x_{t-1}, \ldots, x_{t-k+1})}{\sqrt{\operatorname{var}(x_t | x_{t-1}, \ldots, x_{t-k+1}) \operatorname{var}(x_{t-k} | x_{t-1}, \ldots, x_{t-k+1})}}, \tag{3}$$

where cov refers to the covariance and var refers to the variance and k refers to the lag.

For each signal, we obtained 45 statistical values through a temporal window of 2 minutes with an overlap ratio of 50%. By setting different delays from 2 seconds to 20 seconds in steps of 2 seconds ($k = 20, 40, \cdots, 200$), we extracted 20 linear dependence features. Finally, we obtained a total of $(45 + 20) * 5 = 325$ time domain features.

2.3.2. Frequency Domain Features Extraction. In addition to temporal domain features extracted using statistical methods, we conducted. The discrete Fourier transform to convert data from temporal domain to frequency domain [27]. The formula is defined as

$$F_k = \sum_{j=0}^{n-1} x_j e^{-i2\pi k(i/n)}, \tag{4}$$

where n refers to the length of the signal, i is the sign of complex number.

For each signal, we chose the first 100 amplitudes and phases, respectively. Finally, we obtained a total of $(100 + 100) * 5 = 1000$ frequency domain features.

2.3.3. Dimensionality Reduction. It must be emphasized that driving signals may be interrelated. For instance, the average Pearson correlation coefficient between different signals is shown in Figure 5. Therefore, some of the $325 + 1000 = 1325$ features may be closely related. This redundant information

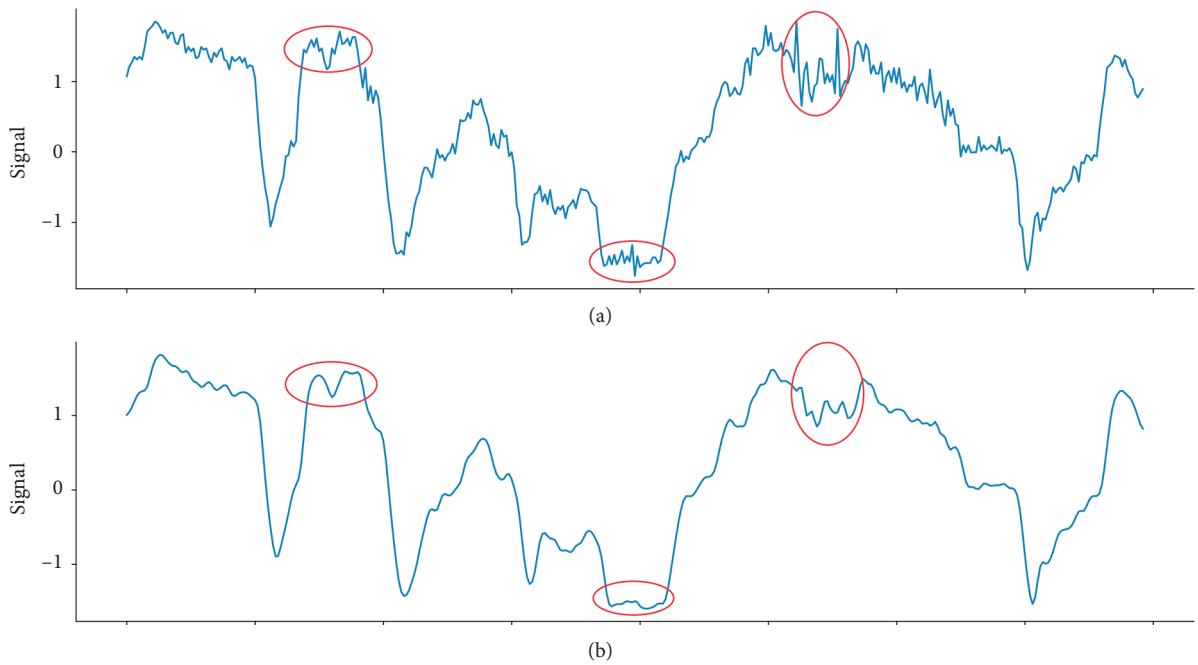


Figure 4: Example of signals before (a) and after (b) filtering.

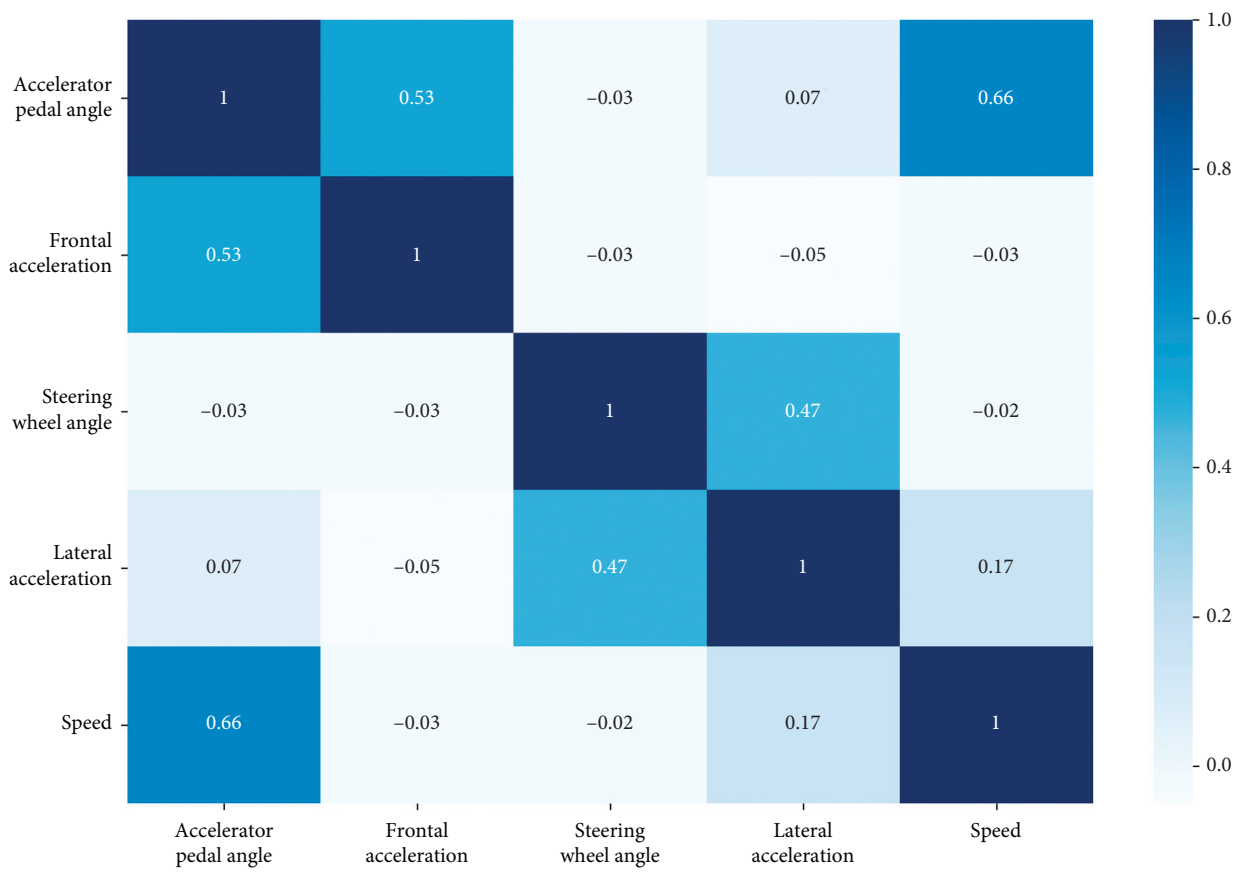


Figure 5: A matrix showing the correlation between the five signals acquired from the CAN bus.

may impact the performance of recognition models, so we need to reduce the redundancy of the feature set.

Since the values of different signals were measured on different scales, in case some important features extracted from signals with small values might be ignored, all features were firstly processed by Z-score normalization. Principal Component Analysis (PCA) was then utilized to reduce the feature dimensions, as it has been demonstrated that PCA could perform much better than other techniques on training sets with small size [28]. To make reconstruction error less than 5%, we reserved 77 principal components as features after dimensionality reduction.

2.3.4. Feature Selection. To get the optimal performance of recognition models, we should find and remove useless features from the above 77 features. In this study, we used the sequential backward selection (SBS) to find the best subset of features that reduced the feature dimension while minimizing the performance loss of recognition model [29], and Algorithm 1 describes the whole process. SBS is a greedy search algorithm that starts from the whole feature set X and sequentially discards the feature $x^{'}$ so as to improve (or minimally worsens) the evaluation measure J. And it stops when the evaluation measure J is not increased or the subset X' is an empty set, which means that all remaining features are useful for the recognition model.

2.4. Model Training. We trained regression models for the recognition of Big Five personality traits. Since there is no evidence showing that a certain machine learning algorithm is the most suitable for identifying personality traits, we investigated the state-of-the-art regression models in this study: linear regression (LR) [30], support vector regression (SVR) [31], and Gaussian process regression (GPR) [32].

LR is a parameter model, whose parameters are estimated by minimizing the mean square error, and makes predictions requires simple matrix multiplication [30]. SVR is an extension of support vector classification, which first maps feature vectors to a higher-dimensional feature space using kernel trick and then makes predictions based only on support vectors [31]. In contrast to the above described algorithms, GPR is a nonparametric kernel-based probabilistic model, with the advantage of automatic tuning of the kernel parameters from the training data by maximizing log marginal likelihood [32].

In this study, we took the linear kernel function for SVR, and the kernel function of the dot-product kernel plus the white kernel for GPR. To evaluate the predictive performance of the models, we considered the root mean squared error (RMSE), the coefficient of determination (R^2), and the Pearson correlation coefficient (r) between predicted scores and self-reported scores of the respective personality traits. Denote C, Γ as a regression function and its corresponding parameters set and $f_i, i = 1, 2, \cdots, n$ as the ith sample's feature set. The RMSE, R^2 and r can be written as

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(C\left(f_i,\Gamma\right)-L_i\right)^2}, \\ R^2 &= 1-\frac{\sum_{i=1}^{n}\left(C\left(f_i,\Gamma\right)-L_i\right)^2}{\sum_{i=1}^{n}\left(C\left(f_i,\Gamma\right)-\overline{L}\right)^2}, \\ r &= \frac{\sum_{i=1}^{n}\left(C\left(f_i,\Gamma\right)-\overline{C\left(f,\Gamma\right)}\right)\left(L_i-\overline{L}\right)}{\sqrt{\sum_{i=1}^{n}\left(C\left(f_i,\Gamma\right)-\overline{C\left(f,\Gamma\right)}\right)^2}\sqrt{\sum_{i=1}^{n}\left(L_i-\overline{L}\right)^2}}, \end{aligned} \tag{5}$$

$C(f_i, \Gamma)$ outputs the predicted score from features f_i and L_i refers the true score of the ith sample. In this work, we applied 10-fold cross validation and averaged performance measures across all folds within a single prediction model.

3. Results

3.1. Demographics and Questionnaire Scores of BFI-44. Of these 92 participants (52 males and 40 females), their ages ranged from 21 to 56 years (mean = 31.84, SD = 7.03), and their driving experience ranged from 0.5 to 33 years (mean = 7.84, SD = 5.96). In terms of education level, the participants reported having the following levels: below university diploma, 2.17% ($n = 2$); university diploma, 51.09% ($n = 47$); and above university diploma, 46.74% ($n = 43$). Descriptive statistics of self-reported personality traits are provided in Table 1. Of the 92 participants who formed the study sample, the personality traits scores between two genders showed no significant difference (openness: $t = 0.62$, $p = 0.53$; conscientiousness: $t = 1.14$, $p = 0.26$; extraversion: $t = -1.16$, $p = 0.25$; agreeableness: $t = -0.08$, $p = 0.45$; neuroticism: $t = -1.38$, $p = 0.17$), which means that gender was not a factor which affects the performance of the Big Five personality traits recognition models in our date set.

3.2. The Recognition of Big Five Personality Traits. After feature selection, the remaining features were different according to regression algorithms. The number of remaining features for LR, SVR, and GPR was shown in Table 2.

The performance of the regression models is presented in Figure 6 and Table 3. The results showed that personality traits can be revealed through driving signals. Specifically, for the five dimensions of personality traits, the best performance occurred with SVR predicting openness (RMSE = 2.47, $R^2 = 0.74$, $r = 0.88$), followed by SVR predicting conscientiousness (RMSE = 2.94, $R^2 = 0.54$, $r = 0.79$), SVR predicting extraversion (RMSE=3.33, $R^2 = 0.45$, $r = 0.75$), SVR predicting agreeableness (RMSE = 3.48, $R^2 = 0.38$, $r = 0.73$), and LR predicting neuroticism (RMSE = 4.23, $R^2 = 0.19$, $r = 0.57$). Furthermore, our results indicated that the performances of different models varied. The results showed that the average

Input:
X: The whole feature set
J: Evaluation measure.
Output:
X': The best subset of features.
$X' = X$,
repeat
$x' = \arg\max_{x \in X}\{J(X' - x)\}$,
$X' = X' - x'$,
until not improvement in J OR $X' = \varnothing$

Algorithm 1: Sequential backward selection.

Table 1: Descriptive statistics of personality variables.

	All		Male		Female	
	M	SD	M	SD	M	SD
Openness	36.61	5.50	36.92	5.31	36.20	5.64
Conscientiousness	34.35	4.81	34.85	4.47	33.70	5.09
Extraversion	27.77	5.29	27.21	5.25	28.50	5.18
Agreeableness	35.16	5.61	34.77	6.29	35.67	4.45
Neuroticism	19.21	5.33	18.54	5.14	20.08	5.38

Table 2: Number of remaining features after feature selection.

	LR	SVR	GPR
Openness	39	39	47
Conscientiousness	23	21	30
Extraversion	31	34	36
Agreeableness	37	45	32
Neuroticism	24	16	27

performance of the SVR model is better than the LR model and GPR model.

4. Discussion

We collected driving signals provided by in-vehicle sensors using CAN bus and trained machine learning models for identifying an individual's Big Five personality traits. Using the time-frequency analysis method, we extracted features from driving signals in the time and frequency domains, respectively, which were used to build personality traits recognition models. For the five personality trait dimensions, the coefficients of determination of the different models were between 0.19 and 0.74, the root mean squared errors were between 2.47 and 4.23, and the correlations between self-reported questionnaire scores and predicted scores were considered medium to strong (0.56–0.88). Our findings demonstrated that driving signals can be used to automatically identify individual personality traits in real-time.

Our results shown the driving signals are a convenient and objective source for measuring individual personality traits. As can be seen from our work, participants only need to drive for less than 10 km before their personality traits can be identified quite precisely. These results were consistent with previous studies showing an association between personality traits and driving behavior [2, 4, 6]. It is worth noting that the effective machine learning models in this current study were built based on low-level features in the time and frequency domains. The high-level features of driving behaviors in this field (e.g., lane switching, tailgating, overtaking, and speeding) are often based on subjective qualitative evaluations [33, 34], which limits the effectiveness of integrating these features into one machine learning model in practice. Although, time-frequency features may not provide much intuitive understanding of individual driving behaviors, they could provide more comprehensive information about driver's personality reflected in driving. Our results demonstrated the validity of building machine learning models to identify self-reported personality traits based on low-level features extracted using the time-frequency analysis.

Modern cars have recently become equipped with several hundred sensors and ECUs, which means we can easily obtain driving signals at minimal cost. Thus, this method to identify personality traits based on driving signals is suitable for the development in-car integration and single-chip embedded systems. Additionally, personalization in the automobile sector is a relatively recent trend to ensure optimal user experience in recent years [35]. Although personalization can be explicitly implemented by providing drivers with system parameters that can be manual tune, the

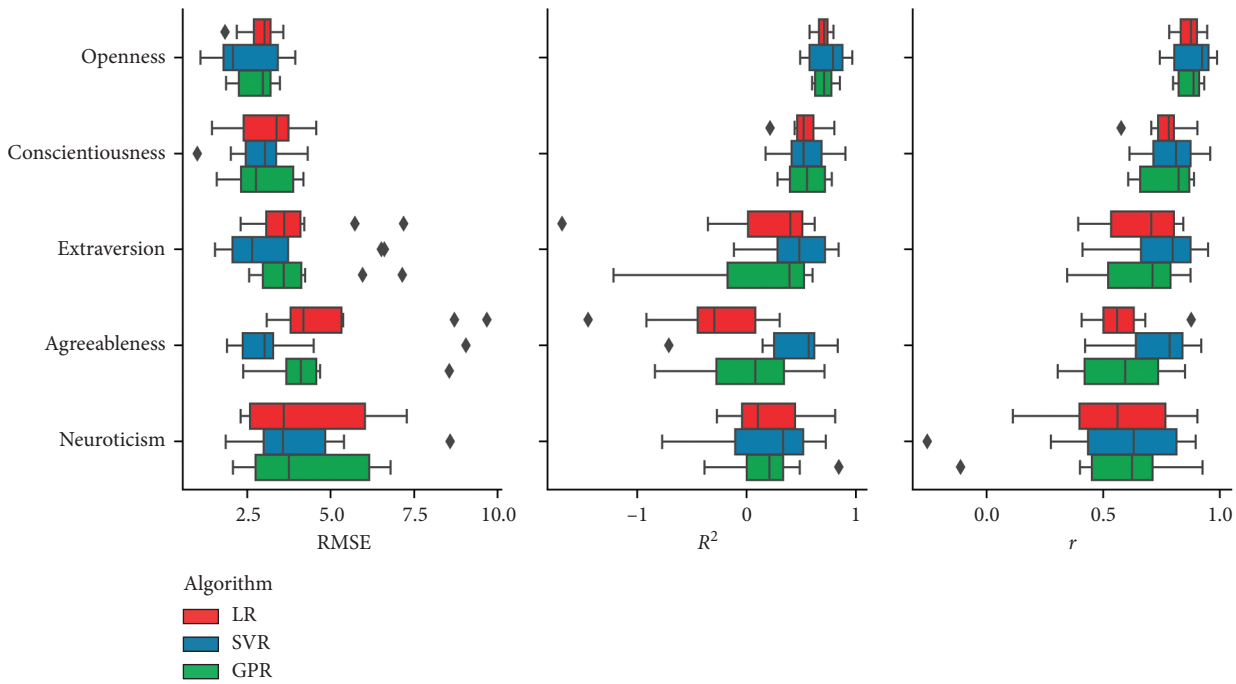


Figure 6: Box and whisker plot of prediction performance measures from 10-fold cross validation.

Table 3: The performance of the regression models.

	LR			SVR			GPR		
	RMSE	R^2	r	RMSE	R^2	r	RMSE	R^2	r
Openness	2.87	0.69	0.87	**2.47**	**0.74**	**0.88**	2.78	0.71	0.87
Conscientiousness	3.07	0.53	0.76	**2.94**	**0.54**	**0.79**	2.93	0.55	0.77
Extraversion	3.95	0.11	0.66	**3.33**	**0.45**	**0.75**	3.96	0.13	0.65
Agreeableness	5.16	−0.32	0.57	**3.48**	**0.38**	**0.73**	4.31	0.02	0.58
Neuroticism	**4.23**	**0.19**	**0.57**	4.12	0.19	0.56	4.26	0.20	0.56

implicit mode that estimates drivers preferences based on observing their behavior not only reduces the tedious and error-prone task of manual tuning, but also satisfies drivers' need for vehicle adaptation through fine-tuning [36]. For example, the "Intelligent Personal Assistant" (IPA) in vehicles is an important feature which offers a way for drivers to interact with their vehicles using their voice [37]. Identifying driver's personality traits by driving behavior and personalizing the IPA dynamically to the current driver will increase the customer experience. Therefore, this method may have potential value of the development of human-centered intelligent driving environments.

As a pilot study, it is appropriate to highlight several limitations. First, in this study personality traits were measured using self-report questionnaires. Although the validity of the questionnaires in accessing personality traits has been well supported in the literature [22], more criteria could be included in future studies. Second, this study's sample population comprised white-collar workers and was not sufficiently large. Therefore, the validity of our model in identifying self-reported personality traits cannot be equated with the effectiveness in populations of individuals with different occupations, education levels, and cultures. Third, the current study built recognition models based on low-level features extracted using the time-frequency analysis, which cannot provide a clear understanding for the relationship between driving behavior and personality. Further research based on intuitively visible high-level features is necessary. Fourth, although our results showed the validity of identifying personality traits using this model, why the performance of models of personality traits in different dimensions is varied remains unclear. The disparity of the accuracies in identifying different dimensions implied that not all the personality-relevant could be equally reflected in driving. For a better understanding of how driving behavior reflects individual personality traits, more future works need to continue from two aspects: first, conducting more experiments, such as driving simulator experiments using fMRI technology [38]; second, explore the relationship between driving behavior and personality traits using more in-depth analysis, such as factor analysis.

Despite those limitations due to the exploratory nature of the study, it suggests the potential in future research on data-driven psychological measurement. Driving signals

have the advantages of being real-time, continuous, nonintrusive, and reliable [39], while requiring him/her finishing a questionnaire frequently and repeatedly is often not acceptable in practice; therefore, this method can measure personality traits in real-time and objectively, which cannot be achieved by a questionnaire. So our recognition model may show advantages in some cases, such as the driver is nonfixed but has a high demand for vehicle adaptation. Moreover, future research can transfer this method to the recognition of other psychological indicators in driving environment, because this method can monitor the continuous change of driver's psychological indicators. Additionally, although technological progress enables increasing automation in vehicles, the current general assumption for designing driving systems, such as driving assistance systems, is that drivers prefer to use systems that adopt a similar driving style to their own [8]. However, there is little empirical evidence to support this assumption. Thus, this method provides a new direction for the research on designing driving assistance systems.

5. Conclusions

This study moved one step forward toward a low-cost, nonintrusive solution for real-time identification of Big Five personality traits, which could be of potential value in the development of in-car integration. Our experiment demonstrated that driving signals provided by in-vehicle sensors using CAN bus can be an objective data source for measuring personality traits, and the predictive machine learning models showed effectiveness in identifying self-reported personality traits. Furthermore, this pilot study indicated a possible direction for further investigation on convenient psychometric methods and provided new perspectives for the development of intelligent driving environments from a human-centered perspective.

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 work was supported by the BMW China Research Project (grant no. 20170321), National Natural Science Foundation of China (grant no. 31700984), and Youth Innovation Promotion Association CAS.

References

[1] L. Mallia, L. Lazuras, C. Violani, and F. Lucidi, "Crash risk and aberrant driving behaviors among bus drivers: the role of personality and attitudes towards traffic safety," *Accident Analysis & Prevention*, vol. 79, pp. 145–151, 2015.

[2] N. J. Starkey and R. B. Isler, "The role of executive function, personality and attitudes to risks in explaining self-reported driving behaviour in adolescent and adult male drivers," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 38, pp. 127–136, 2016.

[3] E. R. Dahlen, R. C. Martin, K. Ragan, and M. M. Kuhlman, "Driving anger, sensation seeking, impulsiveness, and boredom proneness in the prediction of unsafe driving," *Accident Analysis & Prevention*, vol. 37, no. 2, pp. 341–348, 2005.

[4] J. Yang, F. Du, W. Qu, Z. Gong, and X. Sun, "Effects of personality on risky driving behavior and accident involvement for Chinese drivers," *Traffic Injury Prevention*, vol. 14, no. 6, pp. 565–571, 2013.

[5] W. Arthur Jr and D. Doverspike, "Predicting motor vehicle crash involvement from a personality measure and a driving knowledge test," *Journal of Prevention & Intervention in the Community*, vol. 22, no. 1, pp. 35–42, 2001.

[6] B. Shen, W. Qu, Y. Ge, X. Sun, and K. Zhang, "The relationship between personalities and self-report positive driving behavior in a Chinese sample," *PLoS One*, vol. 13, no. 1, 2018.

[7] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[8] M. Hasenjager, M. Heckmann, and H. Wersing, "A survey of personalization for advanced driver assistance systems," *IEEE Transactions on Intelligent Vehicles*, vol. 5, 2019.

[9] O. P. John and S. Srivastava, "The Big Five trait taxonomy: history, measurement, and theoretical perspectives," *Handbook of Personality: Theory and Research*, vol. 2, pp. 102–138, 1999.

[10] S. Soldz and G. E. Vaillant, "The Big Five personality traits and the life course: a 45-year longitudinal study," *Journal of Research in Personality*, vol. 33, no. 2, pp. 208–232, 1999.

[11] G. Wu, F. Chen, X. Pan, M. Xu, and X. Zhu, "Using the visual intervention influence of pavement markings for rutting mitigation-part I: preliminary experiments and field tests," *International Journal of Pavement Engineering*, vol. 20, no. 6, pp. 734–746, 2019.

[12] F. Chen, H. Peng, X. Ma, J. Liang, W. Hao, and X. J. T. Pan, "Examining the safety of trucks under crosswind at bridge-tunnel section: a driving simulator study," *Tunnelling and Underground Space Technology*, vol. 92, Article ID 103034, 2019.

[13] U. Kiencke, S. Dais, and M. Litschel, "Automotive serial controller area network," *SAE Transactions*, vol. 95, no. 2, pp. 823–828, 1986.

[14] Z. Qi, Q. Shi, and H. Zhang, "Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays," vol. 67, no. 7, pp. 5637–5646, 2019.

[15] K. Jiang, H. Zhang, H. R. Karimi, J. Lin, and C. Systems, "Simultaneous input and state estimation for integrated motor-transmission systems in a controller area network environment via an adaptive unscented kalman filter," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, 2018.

[16] On Board Diagnostic Net, "Building adapter for vehicle on-board diagnostic," vol. 9, 2009, https://www.obddiag.net/adapter.html.

[17] A. E. B. El Masri, H. Artail, and H. Akkary, "Toward self-policing: detecting drunk driving behaviors through sampling CAN bus data," in *Proceedings of the 2017 International*

Conference on Electrical and Computing Technologies and Applications, pp. 1–5, Ras Al Khaimah, UAE, January 2018.

[18] A. B. Makar, K. E. McMartin, M. Palese, and T. R. Tephly, "Formate assay in body fluids: application in methanol poisoning," *Biochemical Medicine*, vol. 13, no. 2, pp. 117–126, 1975.

[19] V. Sadhu, T. Misu, and D. Pompili, "Deep multi-task learning for anomalous driving detection using CAN bus scalar sensor data," 2019, http://arxiv.org/abs/1907.00749.

[20] Z. E. A. Elassad, H. Mousannif, H. Al Moatassime, and A. Karkouch, "The application of machine learning techniques for driving behavior analysis: a conceptual framework and a systematic literature review," *Engineering Applications of Artificial Intelligence*, vol. 87, p. 103312, 2020.

[21] P. Wan, C. Wu, Y. Lin, and X. Ma, "Driving anger states detection based on incremental association markov blanket and least square support vector machine," *Discrete Dynamics in Nature and Society*, vol. 2019, Article ID 2745381, 2019.

[22] R. Carciofo, J. Yang, N. Song, F. Du, and K. Zhang, "Psychometric evaluation of Chinese-language 44-item and 10-item big five personality inventories, including correlations with chronotype, mindfulness and mind wandering," *PloS One*, vol. 11, no. 2, 2016.

[23] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 16, no. 2, pp. 34–50, 2016.

[24] T. Lajunen, J. Karola, and H. Summala, "Speed and acceleration as measures of driving style in young male drivers," *Perceptual and Motor Skills*, vol. 85, no. 1, pp. 3–16, 1997.

[25] B. D. Anderson and J. B. Moore, *Optimal Filtering*, Courier Corporation, North Chelmsford, MA, USA, 2012.

[26] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control, John Wiley and Sons*, NJ, USA, Hoboken, 2015.

[27] S. Weinstein and P. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 628–634, 1971.

[28] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[29] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[30] G. A. Seber and A. J. Lee, *Linear Regression Analysis*, John Wiley & Sons, Hoboken, NJ, USA, 2012.

[31] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[32] C. E. Rasmussen, *Gaussian Processes in Machine Learning*, pp. 63–71, MIT Press, Cambridge, MA, USA, 2005.

[33] C. Atombo, C. Wu, M. Zhong, and H. Zhang, "Investigating the motivational factors influencing drivers intentions to unsafe driving behaviours: speeding and overtaking violations," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 43, pp. 104–121, 2016.

[34] R. Fernandes, J. Hatfield, and R. F. Soames Job, "A systematic investigation of the differential predictors for speeding, drink-driving, driving while fatigued, and not wearing a seat belt, among young drivers," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 13, no. 3, pp. 179–196, 2010.

[35] M. Hasenjäger, M. Heckmann, and H. Wersing, "A survey of personalization for advanced driver assistance systems," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 335–344, 2020.

[36] H. Fan and M. S. Poole, "What is personalization? perspectives on the design and implementation of personalization in information systems," *Journal of Organizational Computing and Electronic Commerce*, vol. 16, no. 3-4, pp. 179–202, 2006.

[37] N. Horn, "Hey BMW, now we're talking! bmws are about to get a personality with the company's intelligent personal assistant," 2018, https://www.press.bmwgroup.com/global/article/attachment/T0284429EN/413869.

[38] P. Chen, F. Chen, L. Zhang, X. Ma, and X. J. T. Pan, "Examining the influence of decorated sidewaall in road tunnels using fMRI technology," *Tunnelling and Underground Space Technology*, vol. 99, p. 103362, 2020.

[39] 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.

Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8859891, 11 pages
https://doi.org/10.1155/2020/8859891

Research Article

Development of Driver-Behavior Model Based on WOA-RBM Deep Learning Network

Junhui Liu,[1,2] **Yajuan Jia**,[2] **and Yaya Wang**[2]

[1]*The School of Electro-Mechanical Engineering, Xidian University, The Key Laboratory of Electronic Equipment and Structure Design (Xidian University), Ministry of Education, Xi'an 710071, China*
[2]*The School of Electro Engineering, Xi'an Traffic Engineering Institute, Xi'an 710300, China*

Correspondence should be addressed to Junhui Liu; liujunhui@stu.xidian.edu.cn

Received 2 August 2020; Revised 2 September 2020; Accepted 15 September 2020; Published 29 September 2020

Academic Editor: Petr Dolezel

Human drivers' behavior, which is very difficult to model, is a very complicated stochastic system. To characterize a high-accuracy driver behavior model under different roadway geometries, the paper proposes a new algorithm of driver behavior model based on the whale optimization algorithm-restricted Boltzmann machine (WOA-RBM) method. This method establishes an objective optimization function first, which contains the training of RBM deep learning network based on the real driver behavior data. Second, the optimal training parameters of the restricted Boltzmann machine (RBM) can be obtained through the whale optimization algorithm. Finally, the well-trained model can be used to represent the human drivers' operation effectively. The MATLAB simulation results showed that the driver model can achieve an accuracy of 90%.

1. Introduction

Driver models can be applied to (1) vehicle dynamics [1] including vehicle component design, vehicle dynamics analysis, overall vehicle stability analysis, and design of onboard controls; (2) intelligent transport systems (ITS) [2, 3] including simulation of traffic flow based on the control theory models of driver behavior and modeling drivers' risk taking behavior; (3) driverless vehicle systems [4]; and (4) traffic energy consumption systems [5]. Traffic energy consumption systems are different from the vehicle dynamics simulation. The traffic energy consumption system will be affected by the road, so our research focuses on how the road grade impacts on the driver's behavior characteristics. Driving behaviors, including acceleration behavior, deceleration behavior, and uniform behavior, have impacts on the driving safety [6, 7], vehicle fuel consumption [8–10], and air pollution [11]. Designing a drive cycle of the vehicle requires investigating and collecting the practical driving data, analyzing the experimental data, and establishing the road vehicle driving conditions using relevant mathematical theoretical methods. The vehicle speed of this paper is collected based on the distance, and we also considered how the road grade influences driving speeds in its operating conditions. The operating conditions of the resulting vehicles can be used to determine the vehicle's fuel consumption and the technical development as well as evaluation of new models. It is very important to establish an accurate driver behavior model. The main factors that affect the accuracy of the driver behavior model include road geometry and weather condition [12, 13]. Since the driver behavior is a very complicated stochastic system [14–17], designing driver behavior modeling is a very challenging task.

At present, the main achievements of research on driver behavior modeling technology are as follows. Cai et al. [18] developed a new concept of the driving fingerprint map to represent driving characteristics. Miyajima and Takeda [19] proposed a driver behavior modeling method by using on-road driving data. The method is realized through statistical machine-learning techniques, such as hidden Markov models and deep learning. Angkititrakul et al. [20] proposed a stochastic

driver behavior model based on Gaussian mixture model framework. This proposed method allows adaptation scheme to enhance the model capability to better represent particular driving characteristics of interest. Shi et al. [21] proposed to evaluate driving styles by normalizing driving behavior based on personalized driver modeling. An aggressiveness index is proposed to quantitatively evaluate driving styles in this method, which can be applied to detect abnormal driving behavior. Yamada and Takahashi [22] proposed a driver behavior modeling method based on real traffic data under varying environmental conditions. In this method, the driving speed is assumed to be a function of several factors such as overall traveling schedule, speed, and road surface conditions. Taniguchi et al. [23] proposed an unsupervised learning method, which is established on the basis of the original double articulation analyzer model. This method predicts possible scenarios of driving behavior by segmenting and modeling incoming driving behavior time series data. Okuda et al. [24] proposed a probability-weighted autoregressive exogenous model wherein the multiple autoregressive exogenous models are composed of the probabilistic weighting functions. This model can represent the actual driving behavior. There are plentiful publications on this topic using different optimization approaches, e.g., the instantaneous optimization algorithm [25], wolf pack algorithm [26], and genetic algorithm [27, 28].

However, methods in [19, 20] are very complex and strongly depend on historical data. Shi et al. [21], propose to quantitatively evaluate driving styles by normalizing driving behavior based on personalized driver modeling. The results show that the prediction accuracy of driving behavior modeling will be affected by complex environment. The establishment of methods in [22–24, 29] requires a large amount of actual driving data as measurement data, which is also strongly dependent on historical data.

To solve these problems, the paper presents a new method of driver behavior model based on WOA-RBM. This method establishes an objective function, which contains the training process of RBM based on real driver behavior data. Then, the best training parameters of RBM are obtained through WOA. Finally, the RBM after training based on the best training parameters can be used to build the driver behavior model.

This paper is organized as follows. Section 2 describes the driving data collection. Section 3 presents the process of driving behavior modeling based on WOA-RBM. Section 4 shows the experimental results, which prove that the proposed method in this paper can achieve a better performance. Conclusions are offered in Section 5.

2. Driving Data Collection

In this section, the driving data come from the measurement data along the highway 120 near Manteca, CA, USA. In order to build a highly accurate driver behavior model, more than 2000 different drivers' driving behavior data were collected for each route. The measurement data were collected in Manteca during June 17th–July 28th (six weeks), 2018 [30]. Vehicle speeds are measured at 9 points identified in Figure 1. From the map (Figure 1), this highway is an approximately straight road. When modeling the driver, only the influence of the road grade on the speed of the driver is considered, so the curvature of the road is not displayed. In future research, we will take curvature of the road into building the driver model. The road grade is small, but it still has a great impact on the energy consumption of the vehicle. The energy reduction comes from two reasons. First, if a vehicle has constant speed or accelerates, less energy is consumed by the power plant (the engine and/or the electric motor) to drive the vehicle downhill than on a flat road because the gravity contributes positive work to overcome the negative work by aerodynamic drag and tire rolling resistance. The vehicle's potential energy is partly converted into its kinetic energy. Second, if a vehicle decelerates on a flat road or a downhill, the reduced kinetic energy is wasted by the brake as heat. On the other hand, if the vehicle decelerates on an uphill, part of the decreased kinetic energy is converted into the vehicle's potential energy and less kinetic energy is wasted by the brake. The gained potential energy can later be converted back into kinetic energy during a downhill. In summary, road slope change turns the vehicle's potential energy into an energy buffer to store the kinetic energy. The data are collected in 5-minute interval between 00:00 and 23:55 every day at each measurement point. Vehicle speeds at positions other than the 9 points are estimated by linear interpolation. The studied route stretches along 6.1 km of highway driving. The altitude varies from approximately 9 to 17 m. And the origin altitude is approximately 13 m while the terminal altitude is approximately 17 m. The road is sampled by 305 even steps with the step length $\Delta s = 6100/305 = 20$ m. The recorded speed trajectories for the first week and the altitude of the road are shown in Figures 2 and 3. The drivers generally increase the velocity between 4 and 6 km because this section of the road sets speed limits.

Figure 4 shows the slope information of road environment. When the slope is larger than 0, the road is an uphill road; when the slope is lower than 0, the road is a downhill road; when the slope is equal to 0, the road is a flat road.

The driving data are shown in Table 1.

The data size is 2004×306, the data in each row represent the driving behaviors of 2004 different drivers in the same road section, and the data in each column show the driving behaviors of each driver at different sample points.

3. Driver Behavior Model Based on WOA-RBM

In this paper, a new driver behavior model based on WOA-RBM is proposed, which imitates human driving behavior during real-world driving. This new method is designed based upon the theory of RBM and WOA.

3.1. Deep Learning Network Based on RBM. The deep learning network based on RBM can solve the problem of the multilayer network training, which is also easy to realize [31, 32]. The structure of deep learning neural network based on RBM is shown in Figure 5.

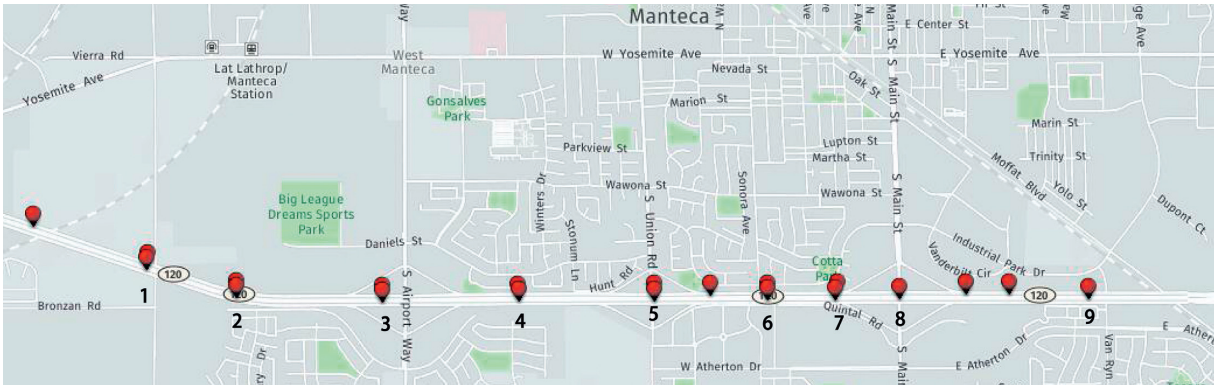


Figure 1: Route of driving data collection.

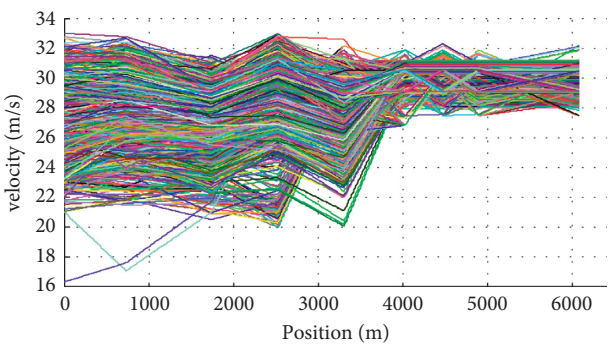


Figure 2: The data of different human driver operations.

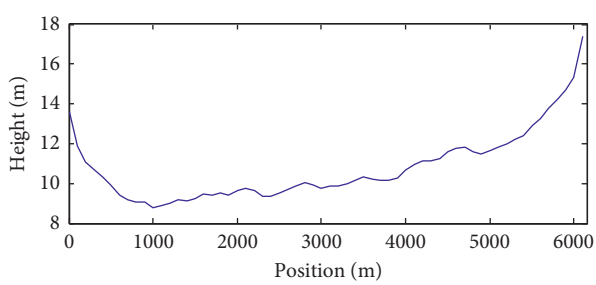


Figure 3: The altitude profile of the road.

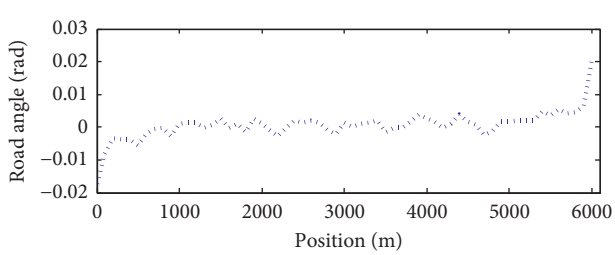


Figure 4: Slope information of road environment.

There are two RBMs and one backpropagation (BP) network in this structure, with a hidden layer and a visible layer in each RBM, both of which are connected through a two-way connection between layers, as shown in Figure 6.

As in references [31, 32], the equation of RBM can be defined as follows:

$$E(v,h) = -\sum_{i\in v} a_i v_i - \sum_{i\in h} b_i h_i - \sum_{i,j} v_i h_i W_{ij}, \tag{1}$$

where $E(v,h)$ is the energy function between the input v vector and the hidden layer output vector h, W_{ij} is a connection weight matrix, v_i is the visible layer, h_i is the hidden layer, a_i is the bias of visible node i, and b_j is the bias of hidden node j.

The probability of each visible v and hidden layer h can be defined as

$$p(v,h) = \frac{1}{\sum_{v,h} e^{-E(v,h)}} e^{-E(v,h)}. \tag{2}$$

The logarithmic gradient of weight in (2) can be calculated as

$$\frac{\partial \log\, p(v,h)}{\partial W_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}, \tag{3}$$

where $\langle v_i h_j \rangle_{\text{data}}$ is the mean value of data and $\langle v_i h_j \rangle_{\text{model}}$ is the mean value of model. Therefore, the learning rules of RBM can be computed as

$$\Delta W_{ij} = \text{Lr} \times \left(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \right), \tag{4}$$

where Lr is the learning rate of RBM.

Similarly, the parameter of indexes in equation (1) can be calculated as follows:

$$\begin{aligned} \Delta a_i &= \text{Lr} \times \left(\langle v_i \rangle_{\text{data}} - \sum_{k=1}^{N} \langle v_i \rangle_k \right), \\ \Delta b_j &= \text{Lr} \times \left(\langle h_j \rangle_{\text{data}} - \sum_{k=1}^{N} \langle h_j \rangle_k \right). \end{aligned} \tag{5}$$

From the above analysis, it is noted that the parameters affecting RBM training performance include the initial value of $\{\text{Lr}, a_i, b_j, W_{ij}\}$, the number of hidden layer h, and the number of visible layer v. Therefore, it is significant to choose the appropriate parameters.

According to the above principle, we assume that the training process of RBM is represented as follows:

Table 1: Segments of the vehicle speeds.

Different drivers	Speeds at sample points (m/s)							
	1	2	3	4	5	6	7	8
1	31.2900	31.2851	31.2803	31.2754	31.2706	31.2657	31.2608	31.2560
2	30.3960	30.3960	30.3960	30.3960	30.3960	30.3960	30.3960	30.3960
3	30.5301	30.5252	30.5204	30.5155	30.5107	30.5058	30.5009	30.4961
4	30.8877	30.8804	30.8731	30.8658	30.8585	30.8531	30.8440	30.8367
5	30.5301	30.5398	30.5495	30.5593	30.5690	30.5787	30.5884	30.5981

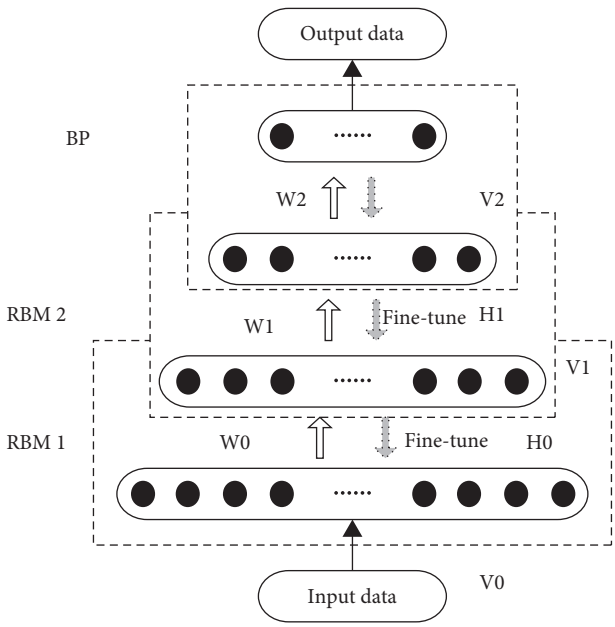


Figure 5: The structure of deep learning neural network based on RBM.

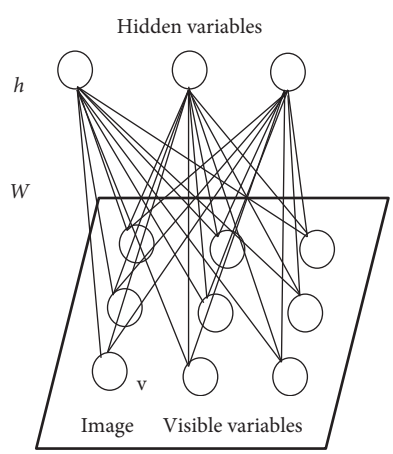


Figure 6: The basic structure of RBM.

$$\text{err} = f_{\text{RBM}}\left(\text{Lr}, v, h, a_i, b_j, W_{ij}\right), \tag{6}$$

where err represents the training error. The smaller the training error is, the closer the driving behavior model similar to the actual driving behavior is.

3.2. Calculation of Optimal RBM Training Parameters. Obviously, the calculation of optimal RBM training parameters is an NP-hard problem. It difficult to calculate the global optimal value directly. To resolve this problem, the paper proposes the whale optimization algorithm (WOA) [33–35]. The main steps of WOA are as follows:

Step 1. Walking and foraging stage: the humpback whale can find the position of food, and the behavior of humpback whale is defined as

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}_t\right|, \tag{7}$$

where $\vec{C}$ is a coefficient vector, $\vec{D}$ is the distance between the whales and food, $\vec{X}_{\text{rand}}$ is the random position, and $\vec{X}_t$ is the current position.

The position of the next moment can be defined as

$$\vec{X}_{t+1} = \vec{X}_{\text{rand}} - \vec{A} \times \vec{D}. \tag{8}$$

The vectors $\vec{A}$ and $\vec{C}$ in equations (7) and (8) can be presented as

$$\begin{aligned} \vec{A} &= 2\vec{a} \cdot \vec{r} - \vec{a}, \\ \vec{C} &= 2 \cdot \vec{r}, \end{aligned} \tag{9}$$

where $\vec{r}$ is a random vector whose values lie in the range of $[0, 1]$ and $\vec{a}$ is the value which is decreased linearly from 2 to 0.

In this paper, the whale population $\vec{X}_t$ is presented as

$$\vec{X}_t = \left[\text{Lr}, v, h, a_i, b_j, W_{ij}\right]. \tag{10}$$

Step 2. Encircling and contracting stage: when the humpback whale finds the target food, the other whales will go to the position of the humpback whale and surround the food. The equation of this stage can be presented as follows:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_t^* - \vec{X}_t\right|,$$
$$\vec{X}_{t+1} = \left|\vec{X}_t^* - \vec{A} \cdot \vec{D}\right|, \tag{11}$$

where $\vec{X}_t^*$ is the vector of random position, which is chose to the current whale population.

Step 3. Spiral predation stage: all the whales will move in a spiral direction of the optimal position of the humpback whale, and then the whales will generate many bubbles to surround the food for predation; the equation of this stage is

$$\vec{X}_{t+1} = \vec{X}_t^* + \vec{D'} \cdot e^{b\vec{l}} \cdot \cos(2\pi\vec{l}), \tag{12}$$

where $\vec{D'} = |\vec{X}_t^* - \vec{X}_t|$ is the distance between the whale and the optimal solutions, b is a shape of logarithmic spiral, and l is a random number in the range of $[-1, 1]$.

Finally, we can define a random value p to distinguish the contraction-bounding stage from the spiral predator. The equation is as follows:

$$\vec{X}_{t+1} = \begin{cases} \vec{X}_t^* + \vec{D} \cdot e^{b\vec{l}} \cdot \cos(2\pi\vec{l}), & p \geq 0.5, \\ \vec{X}^* - \vec{A} \cdot \vec{D}, & p < 0.5. \end{cases} \tag{13}$$

To sum up, the flowchart of WOA is shown in Figure 7.

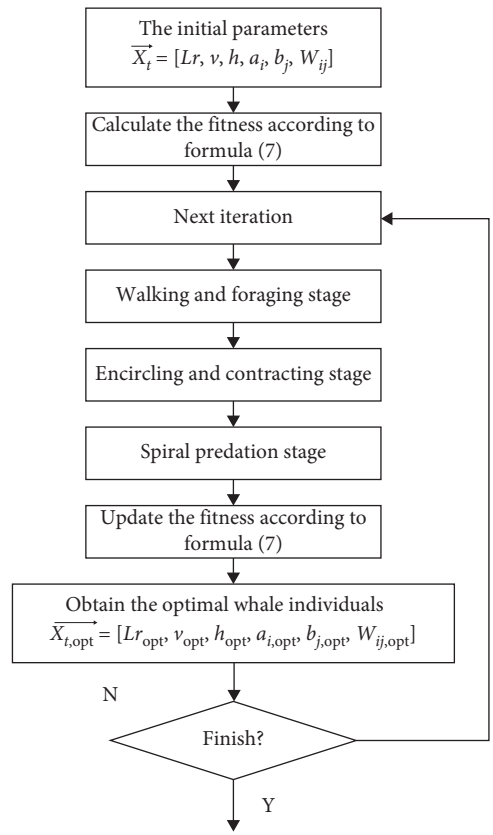


Figure 7: The flowchart of WOA.

3.3. Driver Behavior Model Based on WOA-RBM. Upon the completion of the optimization process, we obtain the optimal parameters:

$$\overrightarrow{X_{\text{opt}}} = \left[\text{Lr}_{\text{opt}}, v_{\text{opt}}, h_{\text{opt}}, a_{i,\text{opt}}, b_{j,\text{opt}}, W_{ij,\text{opt}}\right]. \tag{14}$$

Subsequently, the optimized RBM model can be expressed as

$$E(v,h) = -\sum_{i \in v_{\text{opt}}} a_{i,\text{opt}} v_{i,\text{opt}} - \sum_{i \in h_{\text{opt}}} h_{i,\text{opt}} h_{i,\text{opt}} - \sum_{i,j} v_{i,\text{opt}} h_{i,\text{opt}} W_{ij,\text{opt}}. \tag{15}$$

The learning rules of optimized RBM can be calculated as

$$\begin{aligned} \Delta W_{ij} &= \text{Lr}_{\text{opt}} \times \left(\langle v_{i,\text{opt}} h_{j,\text{opt}} \rangle_{\text{data}} - \langle v_{i,\text{opt}} h_{j,\text{opt}} \rangle_{\text{model}}\right), \\ \Delta a_i &= \text{Lr}_{\text{opt}} \times \left(\langle v_{i,\text{opt}} \rangle_{\text{data}} - \sum_{k=1}^{N} \langle v_{i,\text{opt}} \rangle_k\right), \\ \Delta b_j &= \text{Lr}_{\text{opt}} \times \left(\langle h_{j,\text{opt}} \rangle_{\text{data}} - \sum_{k=1}^{N} \langle h_{j,\text{opt}} \rangle_k\right). \end{aligned} \tag{16}$$

According to the theories introduced above, the driver behavior model based on WOA-RBM is shown in Figure 8. The driver behavior model and the vehicle model constitute a closed-loop control system. The driver behavior model is actually an inverse model of the controlled object (vehicle). In this paper, based on road test data, the learning control method is used to build a driver model through machine learning. Assume that the vehicle speed at a certain position is $V(s)$, and the driver presses down the accelerator pedal or brake pedal which as a consequence will affect the vehicle speed at the next position $V(s+1)$. The expected speed of the next position is $V_e(s+1)$, i.e., the road's speed limit. The collected vehicle speed data are used to train the optimal control parameters which are input into the driver behavior model. At the same time, the road grade, the actual speed of the vehicle, and the deviation between the actual speed and the expected speed are also input into the driver behavior model. Subsequently, the driver behavior model outputs acceleration or deceleration to control the driving of the vehicle.

Figure 8 shows the proposed driver behavior model. In this model, the sampled driver data are used to train the WOA-RBM model. The optimal parameters Lr_{opt}, v_{opt}, h_{opt}, $a_{i,\text{opt}}$, $b_{j,\text{opt}}$, and $W_{ij,\text{opt}}$ of deep learning network are obtained after WOA optimization. The WOA-RBM model after training with the optimal training parameters is the driving behavior model. Finally, the driving behavior model is used in a vehicle control system. In Figure 8, the "road information" specifically refers to the road grade. ΔV indicates the difference between actual speed and desired speed. The whole closed-loop control system output is the vehicle speed. Drive operation is acceleration or deceleration.

4. Experiment

The proposed driving behavior model in this paper is simulated and validated through MATLAB 2017b platform. The performance of the proposed driving behavior model

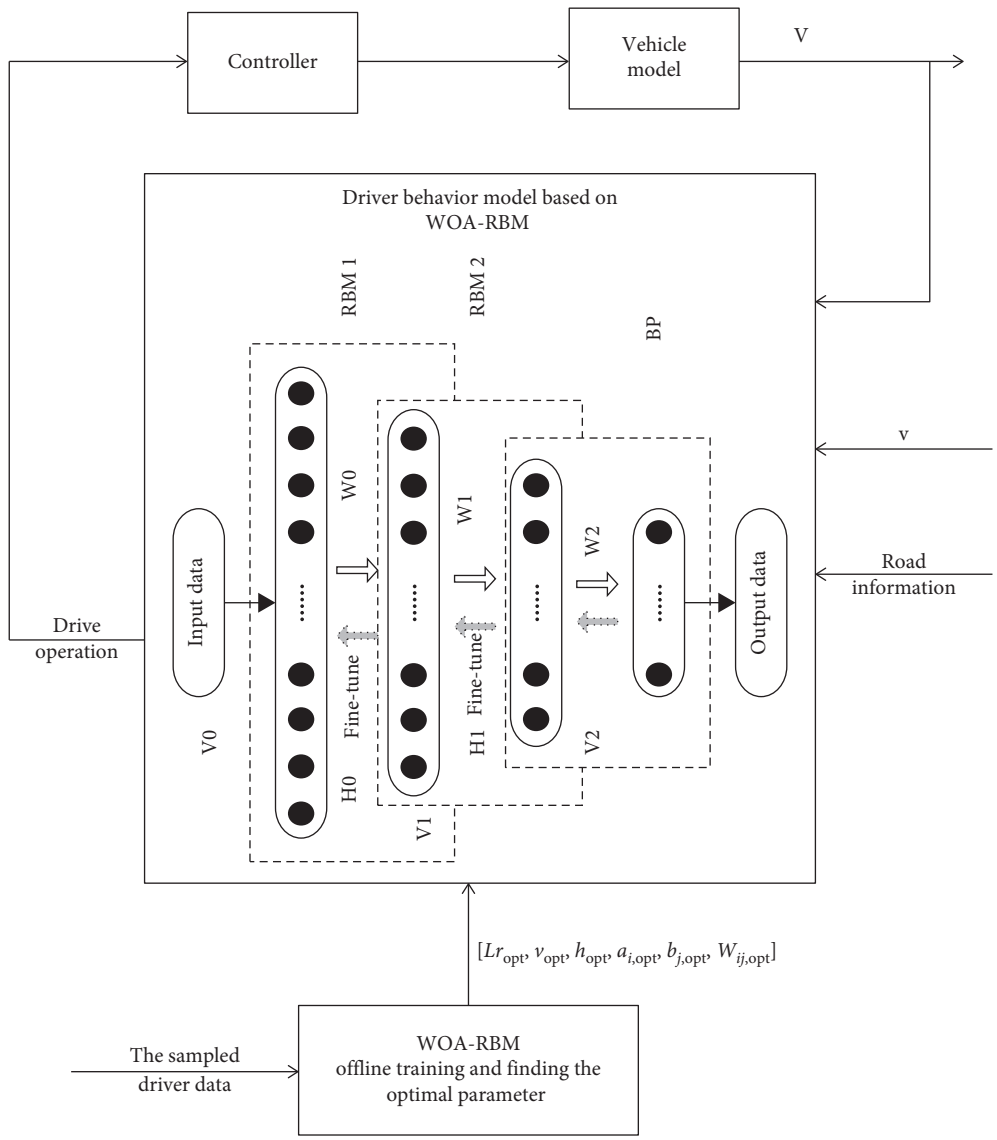


Figure 8: The structure of driver behavior model based on WOA-RBM.

and several other existing driving models are also compared. The configuration of the computer is as follows: Intel i-9700 K processor with 8 cores, CPU frequency of 4 GHz, memory size of 16 G, 64 bit Windows 7 Professional operation system, and Nvidia GeForce GTX 980 graphics card.

4.1. The Parameters. The initial parameters of the proposed driver behavior model are shown in Table 2.

In this experiment, 100 different road sections were collected, and about 2000 drivers' driving data were collected for each route. The driving behavior data of each driver in each route were collected at 306 points.

In this paper, 70 groups are randomly selected as training samples, and the remaining 30 groups are selected as testing samples.

4.2. WOA Optimization Process. The optimization process of WOA is illustrated in Figure 9.

Table 2: The initial parameters.

No.	Variable	Value
1	Number of whale population	50
2	Iteration times	100
3	WOA search space dimension	10
4	Initial learning rate	0.1
5	Batch size	10
6	Number of hidden layers	4
7	Number of visible layers	2
8	Training times	50
9	Number of driver behavior data	2000
10	Length of each driver behavior data	300
11	Average velocity	30 m/s
12	Simulation platform	MATLAB 2017b

Figure 9 shows that when the number of iterations is 9, WOA minimizes the training error of RBM deep learning neural network. The comparison of RBM parameters before and after optimization is shown in Table 3.

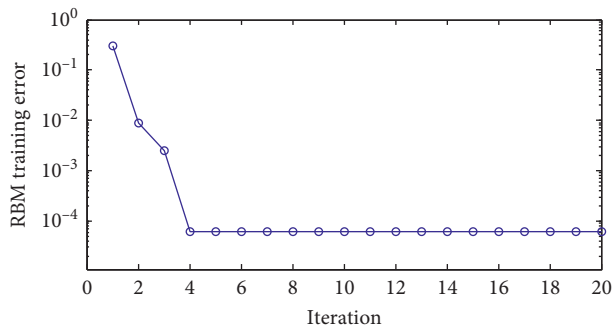


Figure 9: The optimization process of WOA.

Table 3: Simulation parameters.

No.	Parameter	Initial value	Optimal value
1	Lr	1	0.0039
2	v	1	2
3	h	1	2
4	a	0.9134	0.0888
5	b	0.6325	0.1748
6	W	0.0975	0.6883

The simulation results show that with the increase of the number of iterations, the training error of RBM deep learning network gradually reduces. When the number of iterations is 4, WOA minimizes the training error of RBM deep learning neural network, which is within a precision of 10^{-4}. The comparison of RBM parameters before and after optimization is shown in Table 3. After optimized by WOA, the RBM optimal learning rate Lr is 0.0039, the number of visual layers v is 2, the number of hidden layers h is 2, the optimal values of parameters a and b are 0.1748 and 0.6883, respectively, and the optimal initial average of the initial network weights w is 0.088. Under these parameters, the training effect of RBM can obtain the optimal result.

4.3. The Performance of Driver Behavior Model. The information of the road is already shown in Figure 4. The actual driving behavior and the driving behavior predicted by the driver behavior model are simulated in Figure 10.

Figure 10 shows that the six drivers' behaviors from the output of the WOA-RBM-based driver behavior model are consistent with the actual driving behavior. Therefore, the driving behavior model proposed in this paper exhibits high prediction accuracy.

Furthermore, the standard error of estimate (SEE) is used to evaluate the proposed models by more driving behaviors, which is defined as

$$\mathrm{SEE}(i) = \sqrt{\frac{\sum_{n=1}^{N}\left(\mathrm{dat}_{\mathrm{actual}}(i,n) - \mathrm{dat}_{\mathrm{WOA-RBM}}(i,n)\right)^2}{N}}, \tag{17}$$

where $\mathrm{dat}_{\mathrm{actual}}(i,n)$ are the actual driver behavior data, i.e., the vehicle speeds from the $i-$ th driver at the $n-$ th sampled point, and $\mathrm{dat}_{\mathrm{WOA-RBM}}(i,s)$ are the predicted driver behavior data from the $i-$ th driver at the $n-$ th sampled point.

The simulation result of SEE is shown in Figure 11.

Figure 11 shows that the SEE of the driving data and the actual data is obtained by the WOA-RBM model. The simulation results show that the SEE of more than 2000 different drivers is lower than one. Therefore, the driver behavior model based on WOA-RBM can correctly simulate the behavior of the driver.

Through training the 100 driving operation data in different routes, a high level of prediction accuracy is obtained. The corresponding test structure is shown in Figure 12:

From the results of simulation, it can be seen that for 100 different routes, the driving behavior prediction algorithm proposed in this paper can be obtained with lower SEE. Compared with the simulation results in Figure 13, for different routes, the SEE of the driving behavior prediction algorithm proposed in this paper is always close to 0.2. It shows that the accuracy of the algorithm is high.

Overfitting means the training error is very small, while the generalization error is very large. Because the model may be too complex, it "remembers" the training samples, but its generalization error is very high. In the algorithm proposed in this topic, the dropout mechanism is used to prevent overfitting.

4.4. Comparison. To evaluate the performance of the WOA-RBM-based driver behavior model, some other driver behavior models are presented for comparison and analysis, including the model based on database of personal mobility driving [36], stochastic driver pedal behavior model [16], driver behavior modeling using on-road driving data [19], and driver behavior modeling using hidden Markov model based on genetic algorithm [28]. The simulation results of the four methods are shown in Figures 14–16.

The driving behavior predicted by the driver behavior model is shown in Figure 14. The simulation result indicates that the driving data obtained by the proposed WOA-RBM model are closer to the actual driving data.

The SEE value of the outputs of different driving models and the real driving data is calculated according to equation (17). It can be seen that the performance of the WOA-RBM-based driving model proposed in this paper is better than

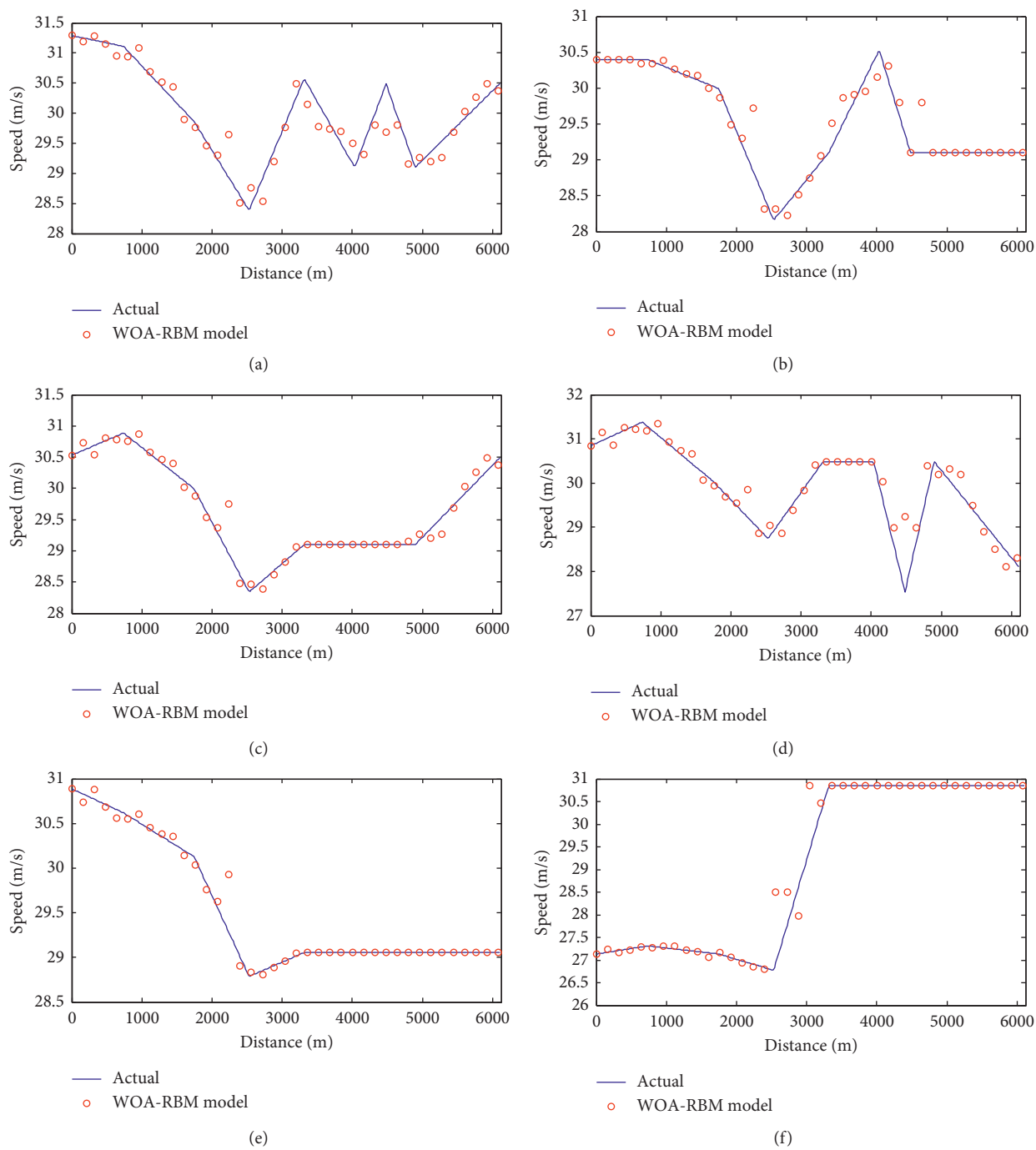


Figure 10: WOA-RBM-based driver behavior model prediction.

that of the driving models proposed in [16, 19, 28, 36]. This is due to the fact that the WOA-RBM-based driving model proposed in this paper can obtain the optimal training parameters of RBM deep learning through WOA and can obtain a model more in line with actual driving data by learning driving training data.

Figure 13 shows that the SEE of the WOA-RBM model is the lowest, while the model in [28] is the largest.

Figure 13 shows the SEE of all the driver models. The driving data are added to verify the influence of different training data, which is shown in Table 4. Comparing the five driving models, the corresponding SEE values are calculated by equation (17) and the WOA-RBM driving behavior prediction model proposed in this paper has the highest prediction performance which has better performance than other algorithms with less training samples.

Table 5 shows the influence of different roads, including smooth road, less undulating road, and large undulating road. The simulation results show that the proposed WOA-RBM model has the best performance among all kinds of

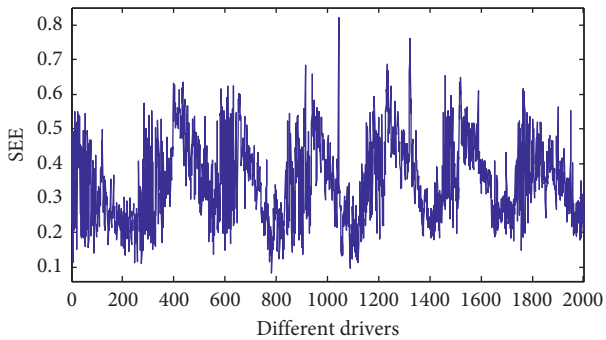


Figure 11: Simulation of SEE.

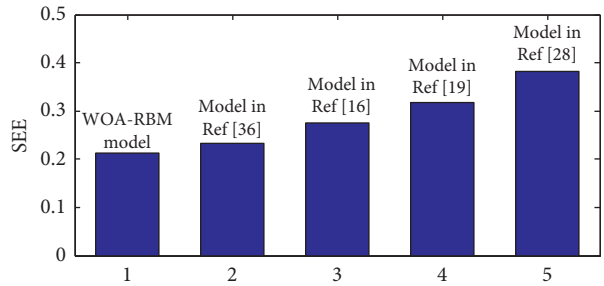


Figure 13: SEE of different driver behavior model predictions.

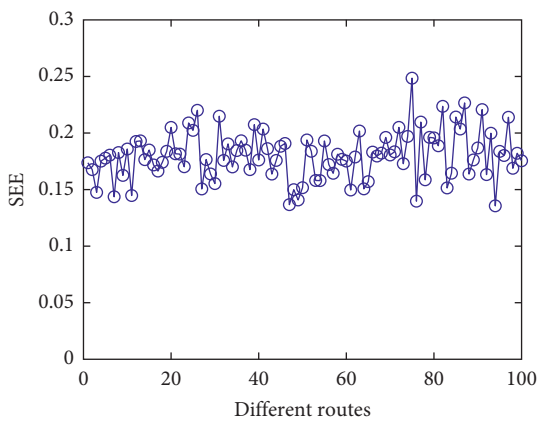


Figure 12: The different routes of SEE.

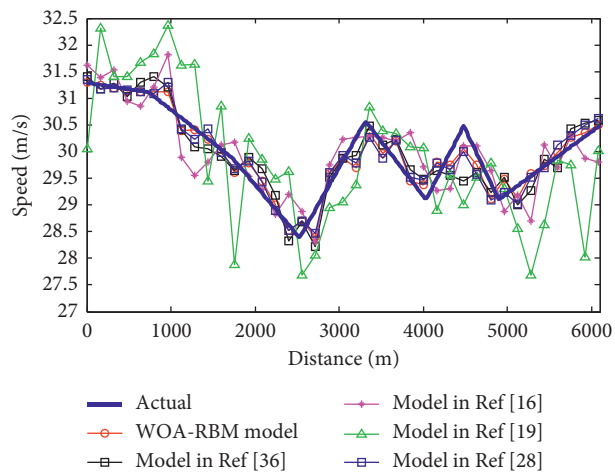


Figure 14: Different driver behavior model predictions.

roads. Several remaining algorithms have larger prediction errors. Especially on road with large undulations, the difference becomes more and more obvious.

On the other hand, the correct and optimal driving behavior can also reduce the energy consumption of the vehicle. In our previous paper [29], the energy consumption model has been discussed. In this paper, the same energy consumption estimation method is used. The energy is calculated as the mechanical energy required for propelling the vehicle. The energy consumption of a vehicle can be estimated from its longitudinal dynamics, as illustrated in Figure 15. By Newton's law, we have the following equation:

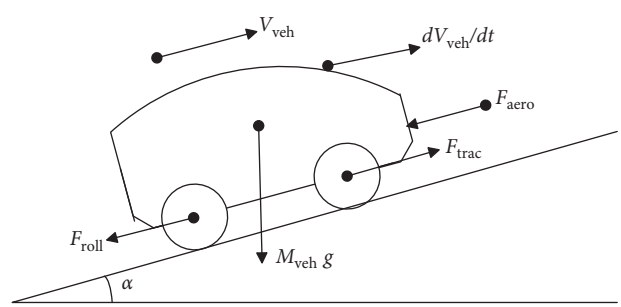


Figure 15: Longitudinal forces on a running vehicle.

$$F_{trac} = M_{veh}a + F_{roll} + F_{aero} + F_{grade} = M_{veh}a + c_{roll}M_{veh}g\cos\alpha + \frac{1}{2}\rho_{air}A_f c_d v_{veh}^2 + M_{veh}g\sin\alpha, \tag{18}$$

where M_{veh} is the vehicle mass, a is the vehicle acceleration, v_{veh} is the vehicle speed, F_{roll} is the rolling friction, F_{aero} is the aerodynamic drag, F_{grade} is the force caused by the road slope, F_{trac} is the traction force generated by the powertrain, g is the gravity acceleration, α is the road grade angle in radiant, c_{roll} is the rolling friction coefficient, ρ_{air} is the density of the ambient air, c_d is the air resistance coefficient, and A_f is the frontal area of the vehicle. Figure 16 compares the performance of energy consumption.

The correct and optimal driving behavior can also reduce the energy consumption of vehicle. Figure 16 shows that the proposed WOA-RBM-based driver behavior model has lower energy consumption than other four models.

When operating a motor vehicle, providing the proper driving operation is essential, as it will reduce energy consumption. This means that frequent acceleration and braking is an improper driving method and will increase energy consumption.

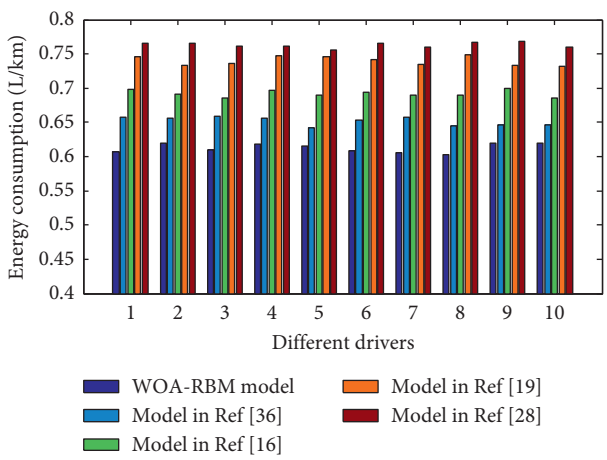


Figure 16: A comparison of energy consumption.

Table 4: The effect of the number of training samples on the performance of the algorithm.

No. of training samples	SEE				
	WOA-RBM	Ref [36]	Ref [16]	Ref [19]	Ref [28]
200	0.205	0.22	0.252	0.311	0.358
100	0.21	0.23	0.26	0.32	0.37
80	0.225	0.264	0.32	0.354	0.412
50	0.246	0.305	0.391	0.415	0.475
20	0.289	0.372	0.445	0.482	0.523

Table 5: The effect of different roads on the performance of the algorithm.

Different roads	SEE					
	WOA-RBM	CD	Ref [36]	Ref [16]	Ref [19]	Ref [28]
Smooth road	0.157	0.178	0.183	0.204	0.224	0.315
Less undulating	0.225	0.245	0.2753	0.283	0.348	0.410
Large undulating	0.310	0.321	0.382	0.412	0.450	0.486

5. Conclusion

In this paper, a new algorithm of driver behavior model based on WOA-RBM deep learning network is proposed. The model establishes an objective function, which contains the training of deep learning network. The RBM after training based on optimal training parameters is used to predict the output action accurately through the road information. In the future, it is of great interest to collect huge amounts of real driving data from different roads and different drivers.

Data Availability

All raw data in this study are available free of charge. Readers who wish to repeat this study can do so through the following link: http://pems.dot.ca.gov/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Scientific Research Program funded by Shaanxi Provincial Education Department (Program no. 20JK0747).

References

[1] M. Plöchl and J. Edelmann, "Driver models in automobile dynamics application," *Vehicle System Dynamics*, vol. 45, no. 7-8, pp. 699–741, 2007.

[2] O. Derbel, T. Peter, H. Zebiri, B. Mourllion, and M. Basset, "Modified intelligent driver model for driver safety and traffic stability improvement," *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 744–749, 2013.

[3] S. Fernandez and T. Ito, "Driver behavior model based on ontology for intelligent transportation systems," in *Proceedings of the 2015 IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA)*, pp. 227–231, Rome, Italy, October 2015.

[4] J. Meech and J. Parreira, "An interactive simulation model of human drivers to study autonomous haulage trucks," *Procedia Computer Science*, vol. 6, pp. 118–123, 2011.

[5] M. Zhang, S. Shi, W. Cheng, Y. Shen, and W. Cao, "Using the amce algorithm to high-efficiently develop vehicle driving cycles with road grade," *IEEE Access*, vol. 7, pp. 160449–160458, 2019.

[6] C. Ma, W. Hao, W. Xiang, and W. Yan, "The impact of aggressive driving behavior on driver-injury severity at highway-rail grade crossings accidents," *Journal of Advanced Transportation*, vol. 2018, Article ID 9841498, 10 pages, 2018.

[7] Y. Ma, Z. Zhang, S. Chen, Y. Yu, and K. Tang, "A comparative study of aggressive driving behavior recognition algorithms based on vehicle motion data," *IEEE Access*, vol. 7, pp. 8028–8038, 2019.

[8] X. Liu, H. Xie, H. Ma, and S. Chen, "The effects of bus driver's behavior on fuel consumption and its evaluation indicator," *Automotive Engineering*, vol. 36, no. 11, pp. 1321–1326, 2014.

[9] D. Hari, C. J. Brace, C. Vagg, J. Poxon, and L. Ash, "Analysis of a driver behaviour improvement tool to reduce fuel consumption," in *Proceedings of the 2012 International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 208–213, 2012.

[10] F. Zheng, J. Li, H. J. van Zuylen, and C. Lu, "Influence of driver characteristics on emissions and fuel consumption," *IET Intelligent Transport Systems*, vol. 13, no. 12, pp. 1770–1779, 2019.

[11] D. W. Wyatt, H. Li, and J. E. Tate, "The impact of road grade on carbon dioxide (CO_2) emission of a passenger vehicle in real-world driving," *Transportation Research Part D: Transport and Environment*, vol. 32, pp. 160–170, 2014.

[12] J. Mclean, "Driver speed behaviour and rural road alignment design," *Traffic Engineering & Control*, vol. 22, no. 4, pp. 208–211, 1981.

[13] S. H. Hamdar, L. Qin, and A. Talebpour, "Weather and road geometry impact on longitudinal driving behavior: exploratory analysis using an empirically supported acceleration modeling framework," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 193–213, 2016.

[14] K. Shaaban and H. Hamad, "Group gap acceptance: a new method to analyze driver behavior and estimate the critical gap at multilane roundabouts," *Journal of Advanced Transportation*, vol. 2018, no. 2, 9 pages, Article ID 1350679, 2018.

[15] A. M. Pérez-Zuriaga, F. J. Camacho-Torregrosa, A. García, and J. M. Campoy-Ungría, "Application of global positioning system and questionnaires data for the study of driver behaviour on two-lane rural roads," *IET Intelligent Transport Systems*, vol. 7, no. 2, pp. 182–189, 2013.

[16] X. Zeng and J. Wang, "A stochastic driver pedal behavior model incorporating road information," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 614–624, 2017.

[17] Liqun, C. Peng, W. Zhen, and M. Huang, "Novel vehicle motion model considering driver behavior for trajectory prediction and driving risk detection," *Transportation Research Record Journal of the Transportation Research Board*, vol. 2434, no. 1, pp. 123–134, 2014.

[18] H. Cai, Z. Hu, Z. Chen, and D. Zhu, "A driving fingerprint map method of driving characteristic representation for driver identification," *IEEE Access*, vol. 6, pp. 71 012–71 019, 2018.

[19] C. Miyajima and K. Takeda, "Driver-behavior modeling using on-road driving data: a new application for behavior signal processing," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 14–21, 2016.

[20] P. Angkititrakul, C. Miyajima, and K. Takeda, "Modeling and adaptation of stochastic driver-behavior model with application to car following," in *Proceeings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 814–819, Baden-Baden, Germany, June 2011.

[21] B. Shi, X. Li, H. Jie, T. Yun, and L. Hui, "Evaluating driving styles by normalizing driving behavior based on personalized driver modeling," *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 45, no. 12, pp. 1502–1508, 2015.

[22] S. Yamada and M. Takahashi, "Analysis of driver behavior based on real expressway data," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 337–342, 2016.

[23] T. Taniguchi, S. Nagasaka, K. Hitomi, K. Takenaka, and T. Bando, "Unsupervised hierarchical modeling of driving behavior and prediction of contextual changing points," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1746–1760, 2015.

[24] H. Okuda, N. Ikami, T. Suzuki, Y. Tazaki, and K. Takeda, "Modeling and analysis of driving behavior based on a probability-weighted arx model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 98–112, 2013.

[25] D. Shi, L. Chu, J. Guo, G. Tian, Y. Feng, and Z. Li, "Energy control strategy of heb based on the instantaneous optimization algorithm," *IEEE Access*, vol. 5, pp. 19 876–19 888, 2017.

[26] L. Zhang, L. Zhang, S. Liu, J. Zhou, and C. Papavassiliou, ""Three–dimensional underwater path planning based on modified wolf pack algorithm," *IEEE Access*, vol. 5, pp. 22783–22 795, 2017.

[27] B. B. Munyazikwiye, H. R. Karimi, and K. G. Robbersmyr, "Optimization of vehicle-to-vehicle frontal crash model based on measured data using genetic algorithm," *IEEE Access*, vol. 5, pp. 3131–3138, 2017.

[28] S. B. Amsalu and A. Homaifar, "Driver behavior modeling near intersections using hidden markov model based on genetic algorithm," in *Proceedings of the 2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pp. 193–200, Singapore, Singapore, August 2016.

[29] J. Liu, L. Feng, and Z. Li, "The optimal road grade design for minimizing ground vehicle energy consumption," *Energies*, vol. 10, no. 5, p. 700, 2017.

[30] Caltrans Performance Measurement System (Pems),"http://pems.dot.ca.gov/, California Department of Transportation.

[31] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[32] Y. Bengio, *Learning Deep Architectures for AI*, Piscataway, NJ, USA, 2009, https://ieeexplore.ieee.org/document/8187120.

[33] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[34] M. Abdel-Basset, M. Gunasekaran, D. El-Shahat, and S. Mirjalili, "A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem," *Future Generation Computer Systems*, vol. 85, 2018.

[35] H. M. Mohammed, S. U. Umar, and T. A. Rashid, "A systematic and meta-analysis survey of whale optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, p. 25, Article ID 8718571, 2019.

[36] K. Inata, P. Raksincharoensak, and M. Nagai, "Driver behavior modeling based on database of personal mobility driving in Urban Area," in *Proceedings of the 2008 International Conference on Control, Automation and Systems*, pp. 2902–2907, Seoul, South Korea, October 2008.