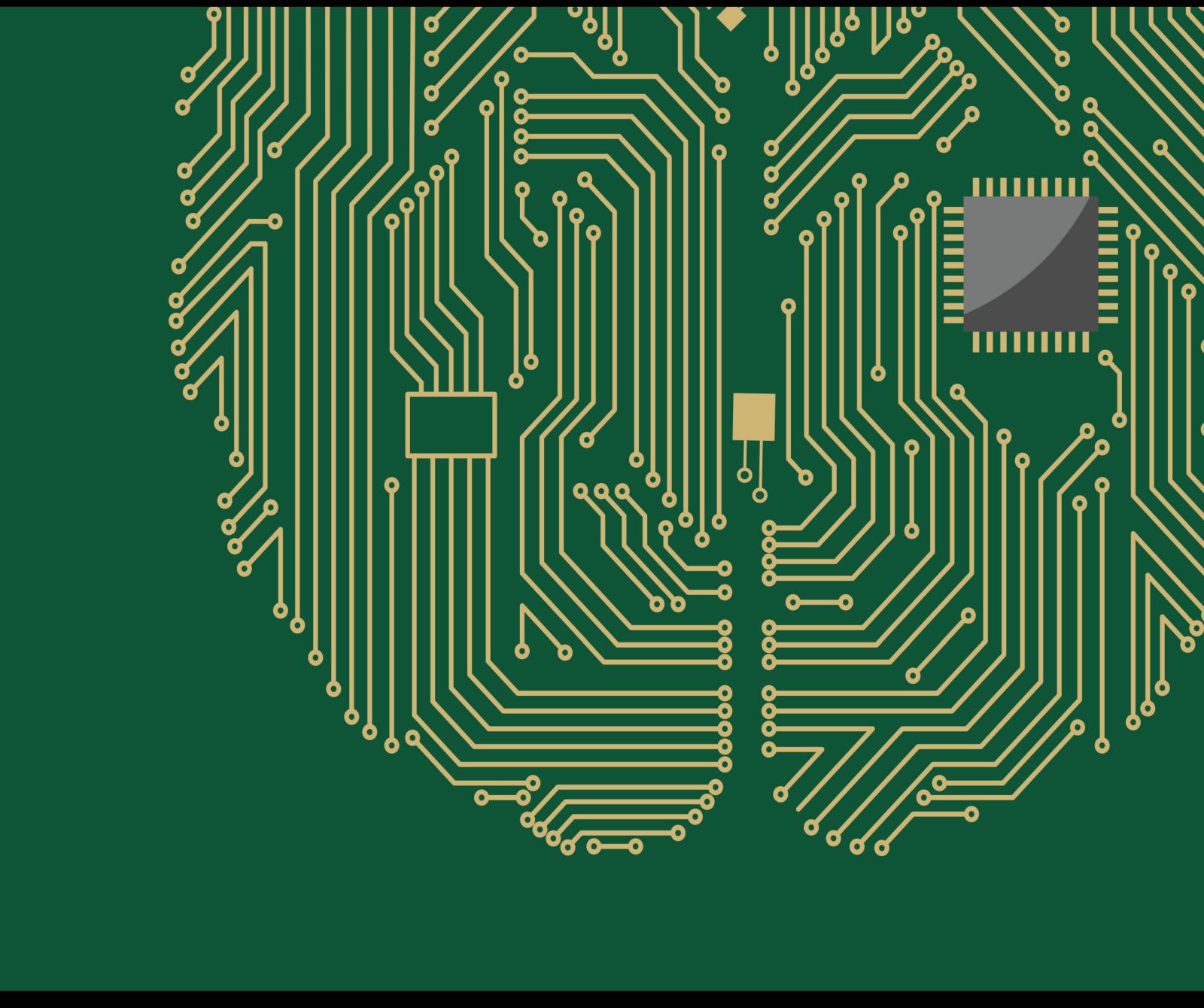
Computational Intelligence and Neuroscience

Neuroevolution: Methods and Applications

Lead Guest Editor: Diego Oliva
Guest Editors: Seyed Jalaleddin Mousavirad, Gerald Schaefer, and Ripon Kumar Chakrabortty



Neuroevolution: Methods and Applications

Computational Intelligence and Neuroscience

Neuroevolution: Methods and Applications

Lead Guest Editor: Diego Oliva
Guest Editors: Seyed Jalaleddin Mousavirad, Gerald Schaefer, and Ripon Kumar Chakrabortty

Contents

Contents

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1452301, 21 pages
https://doi.org/10.1155/2022/1452301

Research Article

An Efficient Hybrid Feature Selection Method Using the Artificial Immune Algorithm for High-Dimensional Data

Yongbin Zhu,[1,2] Tao Li ,[1] and Wenshan Li[1]

[1]*College of Cybersecurity, Sichuan University, Chengdu 610065, China*
[2]*Engineering College, Honghe University, Mengzi 661199, China*

Correspondence should be addressed to Tao Li; taoli.scusc@hotmail.com

Received 11 May 2022; Revised 31 July 2022; Accepted 29 August 2022; Published 13 October 2022

Academic Editor: Seyed Jalaleddin Mousavirad

Feature selection provides the optimal subset of features for data mining models. However, current feature selection methods for high-dimensional data also require a better balance between feature subset quality and computational cost. In this paper, an efficient hybrid feature selection method (HFIA) based on artificial immune algorithm optimization is proposed to solve the feature selection problem of high-dimensional data. The algorithm combines filter algorithms and improves clone selection algorithms to explore the feature space of high-dimensional data. According to the target requirements of feature selection, combined with biological research results, this method introduces the lethal mutation mechanism and the Cauchy operator to improve the search performance of the algorithm. Moreover, the adaptive adjustment factor is introduced in the mutation and update phases of the algorithm. The effective combination of these mechanisms enables the algorithm to obtain a better search ability and lower computational costs. Experimental comparisons with 19 state-of-the-art feature selection methods are conducted on 25 high-dimensional benchmark datasets. The results show that the feature reduction rate for all datasets is above 99%, and the performance improvement for the classifier is between 5% and 48.33%. Compared with the five classical filtering feature selection methods, the computational cost of HFIA is lower than the two of them, and it is far better than these five algorithms in terms of the feature reduction rate and classification accuracy improvement. Compared with the 14 hybrid feature selection methods reported in the latest literature, the average winning rates in terms of classification accuracy, feature reduction rate, and computational cost are 85.83%, 88.33%, and 96.67%, respectively.

1. Introduction

With the continuous in-depth understanding of the research object and the development of data acquisition technology, high-dimensional data has become more and more common. In theory, the more information obtained, the more conducive it is to obtain a more accurate judgment of the object. However, the actual situation may be far from it, because there are often many redundant data irrelevant to the research objectives in these data. They play the role of noise in the pattern recognition model and greatly increase the computational cost of the model. Moreover, in some cases, they will guide the learning process in the model to the generation direction of a weak model, resulting in wrong results. In addition, too many features will not only increase the computational cost in the stage of model training and feature analysis but also lead to the complexity of the model. This can easily lead to the problems of "dimension disaster" and "overfitting" [1]. Reducing the dimension of data is an effective way to solve the classification problem of high-dimensional small sample data [2]. Feature selection technology is widely used to deal with such problems. The goal of feature selection is to select as few features as possible to effectively describe the whole feature space [3]. It can greatly reduce the time of the model in the training stage while maintaining or even improving the classification accuracy.

With the deepening of research and the expansion of application fields, feature selection has been considered to be an important data preprocessing step in the fields of pattern recognition and machine learning [4]. A variety of feature

selection methods have been proposed for different application fields to improve the recognition performance of the model. From a broad point of view, these methods may generally be divided into filter and wrapper methods. In practical application, they have their own advantages and disadvantages. The filter method uses feature correlation criteria to select feature subsets with lower computational costs. The wrapper method uses the classification algorithm to evaluate the quality of the selected features, so as to obtain a higher quality feature subset. In recent years, feature selection methods based on metaheuristics have been the focus of scholars because of their good global search ability [5]. These algorithms include simulated annealing algorithm (SA), genetic algorithm (GA), particle swarm optimization (PSO), and artificial immune algorithm (IA). They all have good global searchability and do not need to provide domain knowledge or prior assumptions about the search space. Moreover, they can effectively deal with complex problems that are difficult to be solved by traditional optimization algorithms without being limited by the nature of the problem. Many literature studies show that feature selection methods based on metaheuristics have excellent performance in solving common feature selection problems. However, with the expansion of search space, especially when the number of features reaches thousands, its calculation cost will increase exponentially [6].

To sum up, the feature selection of high-dimensional data has the following problems. First, the feature subsets obtained by the filter method have low accuracy, which requires artificial analysis of different datasets and selection of specific filter threshold values for them. Second, the metaheuristic-based wrapper method suffers from the problem of high computational cost. In order to solve the above problems, combined with the advantages of the filtering and wrapper method, this paper proposes an efficient hybrid feature selection method based on artificial immune algorithm optimization, namely, HFIA. The algorithm combines the Fisher filtering algorithm and an improved clonal selection algorithm to explore the search space of the optimal feature subset. The Fisher algorithm is an effective filtering feature selection method. It identifies the importance of features by calculating the mean and variance of the distance between and within classes. The artificial immune algorithm is an efficient optimization algorithm that simulates the function of the natural immune system. It is an intelligent algorithm inspired by the principle, function, and model of biological immunity. Based on the traditional evolutionary algorithm, it introduces the mechanism of affinity maturity, cloning, and memory. It has the characteristics of fast convergence speed and strong global optimization ability. It is widely used to solve problems related to optimization and pattern recognition.

According to the target requirements of feature selection, this paper greatly improves the clonal selection algorithm. These improvements include population initialization, mutation strategy, and population update mechanism of antibodies. Combined with the research results of biology, this paper introduces the lethal mutation mechanism and the Cauchy operator to improve the search performance of the algorithm. And different adaptive adjustment factors are introduced in the mutation and update phases of the algorithm. They are used to improve the search speed of the algorithm and enhance the diversity of the population, respectively. The effective combination of these strategies enables the algorithm to obtain better searchability and lower computational costs. The evaluation results on 25 high-dimensional datasets with features ranging from 2000 to 22283 demonstrate the effectiveness of this method. It is compared with 5 classical feature selection methods and 14 hybrid feature selection methods for high-dimensional data reported in the latest literature. The results show that the computational cost of this algorithm is comparable to classical feature selection methods known for their speed. Moreover, it achieves better average classification accuracy than hybrid feature selection methods reported in the latest literature with the smallest number of optimal feature subsets. The comparative experimental results fully demonstrate the progressiveness of the algorithm.

The rest of this paper is organized as follows. Firstly, the feature selection methods and their related domain knowledge are summarized in Section 2. Secondly, the implementation details of the proposed method are described in Section 3. Then, Section 4 provides a detailed description of the experimental datasets, the evaluation metrics, and the algorithm parameter settings. The experimental results are analyzed and discussed in Section 5. Finally, Section 6 gives the conclusion.

2. Preliminaries and Related Work

In this section, feature selection techniques based on metaheuristic algorithms are reviewed first. And the problems of this kind of method in feature selection in high-dimensional data space are proposed. Secondly, the related work of current hybrid feature selection methods is summarized. Finally, the principle of the immune clonal selection algorithm is introduced.

2.1. Feature Selection Based on Metaheuristic Algorithms. Feature selection can filter out the best feature subset that can represent the whole dataset by removing irrelevant or unnecessary features. It is considered to be one of the most critical and challenging problems in machine learning. It is widely used to solve the problem of dimension reduction of datasets in different fields, such as the best gene screening in biomedicine [7], the hot topic recognition in text mining [8], and the best visual content pixel and color selection in image analysis [9]. These algorithms are mainly divided into the filter and wrapper method. The filter method focuses on the internal relationship of data. They are usually not directly related to learning algorithms or classification algorithms. They use different correlation criteria of features to select the optimal feature subset. Therefore, most of them have the advantage of low computing costs. The wrapper method focuses on the interaction results between different feature combinations and classifiers. Compared with the former, these methods need to pay higher computational costs, but

they can provide more accurate results. The generation of the optimal subset of the wrapper method is based on a specific search strategy. These search technologies are mainly divided into three categories. They are the exponential selection strategy, sequential selection strategy, and random selection strategy [10]. In recent years, the metaheuristic algorithms have been widely used in the problem of selecting optimal subsets of feature spaces [5].

They are derived from the heuristic algorithm and are also the product of the combination of the random algorithm and the local search algorithm. They have a good global search ability, and there is no need to provide domain knowledge or advance assumptions about the search space. A metaheuristic algorithm is also called an intelligent optimization algorithm. They are based on the mechanism of computational intelligence to solve some complex optimization problems. The solution obtained by this kind of algorithm is called the optimal solution or satisfactory solution. They have the advantages of simple concepts and easy implementation. In addition, they also have the features of flexibility and intuition and can be modified according to the specific problems that are to be solved. Moreover, they also have the significant advantage of preventing the algorithm from falling into the local optimal solution due to premature convergence, so they can effectively explore the whole problem space. Because of these advantages, the metaheuristic algorithm has attracted extensive attention from researchers. Metaheuristic-based optimization algorithms have also been successfully applied to various engineering and scientific research optimization problems, such as job scheduling, transportation management, vehicle path planning and facility location in industrial engineering, bridge structure and architectural design in civil engineering, radar design and networking in communication engineering, classification, prediction, clustering, and system modeling in data mining.

Feature selection methods are usually implemented by searching the solution space with the goal of maximizing the correlation with the target class and minimizing the redundancy of the selected features [11]. Although this goal can be achieved through the simplest exhaustive search strategy. However, its calculation cost is unacceptable. This is even more unrealistic for large-scale datasets. Since it is too expensive to evaluate all possible feature subsets, a method that is acceptable in terms of computational complexity needs to be used to find suitable feature subsets [12]. The metaheuristic algorithm provides an effective way to solve this kind of problem. It can find a satisfactory near-ideal solution in an acceptable time, although this is not the only optimal solution [13].

Most metaheuristics start by generating several random initial solutions and then evaluating the resulting set of solutions using a fitness function [14]. The approximate optimal solution is searched by continuous loop iteration until one of the termination conditions is satisfied. In addition, people always want to get better results from machine learning models. The strategy of adding different optimization objectives to the fitness function of the feature selection problem emerges as the times require. Using a multi-objective optimization strategy to model the feature selection problem, a set of nondominated feature subsets can be obtained. This solution can meet various requirements in practical applications when the number of features is not too large. However, as the data dimension continues to expand, especially when the number of features reaches thousands, the computational cost will increase exponentially. Therefore, it is urgent to find a more efficient solution to the feature selection problem in high-dimensional data space.

2.2. Hybrid Feature Selection Approaches. Because of the particularity of high-dimensional data space, it is difficult to provide a satisfactory solution whether it is a filter method, a wrapper method, or any other single feature selection method. In recent years, hybrid algorithms have received extensive attention in solving optimization problems. Hybrid algorithms are those that combine different algorithms and develop a new or improved algorithm to solve more complex optimization problems.

In the problem of feature selection, a variety of hybrid algorithms have been used to solve the problem of selecting the optimal feature subset of high-dimensional data. The hybrid feature selection method combines the advantages of different methods. The hybrid scheme of multiple algorithms greatly increases the probability of finding the optimal solution efficiently and quickly. Moreover, some hybrid algorithms combine the best characteristics of different algorithms to develop new algorithms. Therefore, the hybrid algorithm can greatly reduce the search space of the optimal feature subset. In this case, the hybrid algorithm based on a metaheuristic algorithm reduces the possibility of falling into the local optimal solution. This is because when a large number of noisy data are removed, they can better avoid premature convergence and explore the whole data space more effectively. In most cases, the hybrid algorithm can obtain the optimal solution with better quality. Moreover, they can make a better trade-off between search quality and development quality of the algorithm. Therefore, compared with the single feature selection method, the hybrid method has a better application value.

So far, a variety of hybrid feature selection methods have been proposed. Combining global search and local search methods, Liu et al. [15] proposed a hybrid feature selection method based on a genetic algorithm and embedded regularization. Lu et al. [16] proposed a hybrid feature selection algorithm that combines mutual information maximization and an adaptive genetic algorithm. The algorithm first removes a large number of redundant features by maximizing mutual information and then searches for the optimal feature subset by an adaptive genetic algorithm. Ma et al. [17] proposed a two-stage hybrid ant colony algorithm for high-dimensional feature selection. It uses the interval strategy to determine the size of the optimal feature subset searched in the additional stage. This method helps to reduce the complexity of the algorithm and avoid the algorithm falling into local optimization. Huang et al. [18] designed a two-stage hybrid feature selection algorithm. This method combines the binary state transition algorithm and the

ReliefF algorithm and shows a good performance on low-dimensional data with the help of new operators. In addition, Yan et al. [19] proposed a hybrid feature selection algorithm combining simulated annealing and an improved coral reef optimization algorithm. The algorithm is used to solve the feature selection problem of high-dimensional biomedical datasets. Hussain et al. [20] proposed a hybrid optimization method integrating the sine-cosine algorithm into Harris hawks. The method is used to solve the problems of numerical optimization and feature selection. Song et al. [21] proposed a hybrid feature selection based on correlation-guided clustering and particle swarm optimization, which is used to solve the feature selection problem of high-dimensional data. Xiao et al. [22] proposed a hybrid feature selection method that fuses multiple algorithms. The method first filters the feature space by combining the k-means clustering algorithm and the signal-to-noise ratio ranking method, and then combines the cellular learning automaton with the ant colony optimization as a wrapper method to apply to the reduced dataset. Chen et al. [23] proposed a hybrid feature selection method based on evolutionary multitasking. The method first uses the ReliefF algorithm to calculate feature weights and then searches for the optimal feature subset through PSO. Wan et al. [24] proposed a hybrid feature selection method that combines neighborhood rough sets and conditional mutual information. The method first uses various neighborhood information to measure and redefine the correlation of features and then obtains the optimal feature subset through the interactive feature selection algorithm (NCMI_IFS) based on neighborhood conditional mutual information. Song et al. [25] proposed a hybrid feature selection algorithm based on surrogate sample-assisted particle swarm optimization (SS-PSO). The method first uses a cooperative feature clustering mechanism to divide the feature space and then uses PSO to search different feature spaces to obtain the optimal feature subset.

These hybrid methods can overcome the shortcomings of traditional methods in solving the feature selection of high-dimensional data to a certain extent. Compared with the classical filter feature selection method and other methods based on an intelligent algorithm, they can reduce feature redundancy and computational cost to a greater extent. Moreover, they have a better performance in improving the classification performance of the classifier. However, they still have the problem that the classifier performance is not ideal due to too much noise in the selected feature subset to varying degrees. In addition, most algorithms also have the problem of high computational cost due to the complexity of the algorithm itself or insufficient optimization.

2.3. Artificial Immune Optimization Algorithm. The artificial immune algorithm is an intelligent computing method designed to solve complex optimization problems. It is also a metaheuristic algorithm that simulates the operation mechanism of the biological immune system. Since Burnet [26] fully elaborated the principle of clone selection in 1959, the algorithm has been generally recognized by the immunology community. This theory states that B cells with a high affinity for antigens in organisms are retained by the immune system and have the characteristics of clonal proliferation, and these proliferating cells will differentiate into two types of cells with different functions. Some are memory cells that function as antigen markers. Others are plasma cells that destroy antigens, known as antibodies. The theory of clonal selection is used to explain the characteristics of immune responses to antigenic stimulation. The core idea is to select only those cells that can recognize antigens for cloning and proliferation. It describes the properties of the acquired immunity of the biological immune system. The clonal selection mechanism corresponds to the process of affinity maturation of immune cells. That is, under the action of this mechanism, immune cells with lower affinity to antigens undergo a process of "maturing" by gradually increasing their affinity after undergoing clonal proliferation and mutation. During this process, mutations in cloned individuals are inversely proportional to antigen affinity. The production of antibodies is the learning process of the immune system.

Based on clonal selection theory, de Castro and Von Zuben proposed a famous clonal selection algorithm (CLONALG, also known as CSA) in 2000 [27]. It is pointed out in literature that the algorithm is mainly composed of population initialization, clone selection, clone proliferation, hypermutation, and population renewal. Among the important features of the clonal selection algorithm, the hypermutation is an important part. It is the basic guarantee to realize the diversity of algorithms, but the choice is the premise.

The basic clonal selection algorithm consists of the following steps [28]. Each time these steps are performed, a new generation of immune cells will be generated.

① Antibody initialization: generate a set P of candidate solutions.

② Affinity evaluation: calculate the affinity of each antibody in the antibody pool.

③ Selection and cloning: select n antibodies with the highest affinity, and clone these n antibodies in proportion to their affinity with the antigen to form a clone group C.

④ Hypermutation: the clone population is proposed to hypermutation operation, and a mature antibody population D is generated.

⑤ Population update: D is reselected to form a memory cell set M. d new antibodies are generated to replace the lower affinity antibodies in P.

⑥ Repeat steps 2–5 until the termination condition is met.

In recent ten years, the clonal selection algorithm has attracted the attention of many researchers because of its good global optimization ability and convergence performance. Accordingly, the clonal selection algorithm has evolved into many variants and applied to different research

fields. Shang et al. [29] improved the basic CSA in terms of population initialization, clonal selection method, and population update, so as to obtain better convergence when solving multi-objective optimization problems. Dai et al. [30] proposed a clonal selection algorithm based on the bidirectional quantum crossover. In this method, the bidirectional quantum crossover mechanism in quantum jump theory is used to replace the hypermutation operation to realize the information exchange between antibodies. And to improve the search performance of the CSA algorithm. Xu et al. [31] adopted the degradation identification (DR) method to evaluate the suboptimal solution to be eliminated and thus improved the computational efficiency of the CSA algorithm in dealing with complex engineering multimodal optimization problems. Yan et al. [32] improved the clonal selection algorithm and successfully applied it to solve the nonlinear optimization problem in AVO elastic parameter inversion. Luo et al. [33] improved the basic clone selection algorithm to solve the global optimization problem in dynamic multimodal optimization.

3. Proposed HFIA

In this paper, a hybrid feature selection method (HFIA) combining the filter feature selection method with a multi-objective artificial immune algorithm is proposed. This method effectively combines the advantages of the Fisher filtering algorithm and the improved clonal selection algorithm. According to the target requirements of feature selection in high-dimensional data, this method greatly improves the initialization and mutation strategy of antibody population of the clonal selection algorithm. In this section, the implementation details of the HFIA algorithm and several improvements to the clonal selection algorithm will be described in detail. These descriptions include the following aspects: antibody coding mode, affinity evaluation operator, population initialization mode, mutation strategy, and update mechanism of mature antibodies. Finally, the pseudo-code of the algorithm is given at the end of this section, and the symbols used in the algorithm are also explained.

3.1. Solution Encoding. For feature selection methods based on metaheuristic algorithms, binary coding strategy is mostly used to represent feature space. This is because binary vectors can not only easily represent subsets of features but also simplify the operation of the algorithm. Therefore, for feature selection problems, binary encoding is usually adopted to represent the individuals in the solution. In this paper, we also adopt the binary coding strategy. Moreover, this can also make better use of the advantages of the algorithm itself.

In this paper, the encoding is a binary vector of length n. Each bit corresponds to a feature, and n is the size of the feature space. The code uses a value of "0" or "1" to characterize whether the feature at that location is included in the feature subset. A value of "1" indicates that the feature at this location is selected, otherwise it is not selected.

3.2. Evaluating the Fitness. In this paper, a multi-objective optimization strategy is used to model the feature selection problem. The purpose is to obtain a subset of features with a smaller number of features while striving to achieve higher classification accuracy. Therefore, for the feature selection problem, we have two optimization objectives, which are classification accuracy and the number of feature subsets. According to the multi-objective optimization decision-making model introduced above, this paper constructs the following formula (1) as the fitness function. The construction method of the fitness function is also widely used in other literature [34–37] to evaluate the quality of feature subsets.

$$fitness = \omega \times E_r + (1 - \omega) \times \left(\frac{p}{q}\right). \tag{1}$$

Among them, $\omega \in (0, 1)$ is a given real number. In most literature, ω is usually set to 0.9 [38]. E_r is the classification error rate. It is obtained by evaluating this subset of features by an evaluator (usually a classifier). q is the total number of features in the dataset, while p is the number of selected features in the feature subset. In this paper, we use K-nearest neighbor (KNN) as a classification estimator for feature subsets, where $k = 5$.

3.3. Initial Population Generation. In the basic clonal selection algorithm, the initial population is generated by random distribution. In fact, many biological phenomena appear in the form of probability distribution of continuous random variables. In addition, the probability distribution of many random variables takes the normal distribution as its limit distribution under certain conditions. The normal distribution is also known as the Gaussian distribution. The Gaussian distribution is a very important probability distribution in many fields such as mathematics, physics, and engineering. In evolutionary computing, Gaussian distribution is often used in the population mutation link of evolutionary algorithms. Extensive literature studies have shown that large mutational situations in populations are less likely when a Gaussian distribution is used. This may cause the problem of insufficient algorithm diversity. This increases the risk of the search algorithm falling into a local optimum and reduces the convergence speed of the algorithm. Cauchy distribution is another continuous probability distribution function. Compared with Gaussian distribution, its attenuation speed is slower and allows a larger mutation step. This greatly increases the possibility of the algorithm jumping out of the local optimum. Moreover, it has been reported that even in terms of the diversity of methods, the Cauchy distribution is better than the Gaussian distribution in the search process of the evolutionary algorithm [39–41]. The probability density function of the one-dimensional Cauchy distribution is shown in the following formula:

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma\left[1 + \left(x - x_0/\gamma\right)^2\right]} = \frac{1}{\pi}\left[\frac{\gamma}{\left(x - x_0\right)^2 + \gamma^2}\right]. \tag{2}$$

The Cauchy distribution has two parameters, x_0 and γ. x_0 is the position parameter and γ is the scale parameter. They determine the shape of the Cauchy distribution. If the value of γ is larger, the peak height of the probability density function will be smaller and the width will be larger. Conversely, if the value of γ is small, the peak height of the probability density function will be higher, and the peak width will be smaller. When $\gamma = 1$ and $x_0 = 0$, it is called the standard Cauchy distribution. Its probability density function is shown in the following formula:

$$f(x; 0, 1) = \frac{1}{\pi\left(1 + x^2\right)}. \tag{3}$$

Its corresponding cumulative distribution function is shown in the following formula:

$$F(x; x_0, \gamma) = \frac{1}{\pi}\arctan\left(\frac{x - x_0}{\gamma}\right) + 0.5. \tag{4}$$

When the parameters are the same, the probability density functions of the Cauchy and Gaussian distributions are shown in Figure 1. The following conclusions can be drawn intuitively from the figure. Compared with the Gaussian distribution, the Cauchy distribution has a slower decay rate and a larger range of values.

Therefore, in order to obtain the optimal feature subset more quickly, the Cauchy distribution will be applied to the initialization, mutation, and update stages of the population in this paper. At the same time, this is also to reduce the computational cost of the algorithm. The initial population space is generated using the standard Cauchy distribution, which is then transformed into a feature code for antibodies. The standard Cauchy distribution function is shown in formula (3). The outline of the population initialization algorithm is given in Algorithm 1. Firstly, the algorithm generates the real initial population space through the standard Cauchy distribution function. Then, according to the threshold value η, the real number bits in the initial population space are converted into binary bits that can represent the feature code. In this paper, the value of η is -0.2. That is, when $\eta > -0.2$, the locus of the antibody is assigned with a value of 1, otherwise, it is 0.

3.4. Mutation and Update Strategy. The clonal selection algorithm introduces the mutation theory of organism cells to promote the proliferation and evolution of individuals in the population. The mutation mechanism plays an important role in the operation steps of the clone selection algorithm. It gives the algorithm the capability of local random search. At the same time, it also has the function of maintaining the diversity of the population and preventing the phenomenon of premature convergence of the algorithm. Mutations of individuals in a population are carried out at randomly selected loci. Its fundamental purpose is to make the population more diverse. Hypermutation is an important mechanism for the biological immune system to recognize external invasion. It obtains a higher affinity for the antigen through the mutation mechanism of the antibody gene. Due to the semi-blindness of the clone selection algorithm in the search problem, scholars have proposed various mutation strategies to improve the algorithm. These mutation strategies usually have differences for different problem domains.

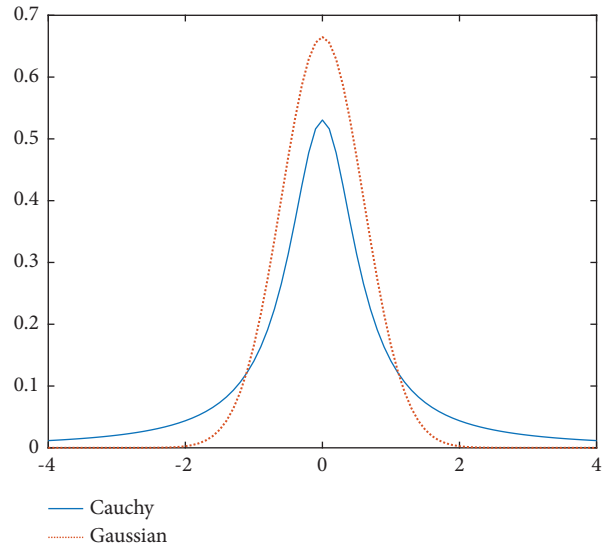


Figure 1: Comparison of probability density functions between Cauchy distribution and Gaussian distribution.

The fundamental purpose of feature selection is to find a better subset of features to represent the entire feature space, that is, to find a subset with less feature redundancy and higher classification accuracy. Based on this, this paper is inspired by the phenomenon of lethal mutation in gene mutation theory and performs lethal mutation operations on elite antibodies in the population. In this way, the algorithm is accelerated to search in the direction of a smaller number of feature subsets. From a biological point of view, although lethal mutations are detrimental to lethal individuals, they are beneficial for maintaining the heterozygous state of the population. The experimental results show that it can make the algorithm obtain better one-way search ability in solving the feature selection problem of high-dimensional data. Therefore, a feature subset with less feature redundancy can be obtained while ensuring classification accuracy. Moreover, it can also reduce the computational cost of the algorithm to a greater extent.

Furthermore, to steer the mutation process in the direction required by the problem domain, an adaptive linear acceleration factor $\delta \in (-0.5, 0.5]$ was added to the mutation process of the elite antibodies. Its effect is to accelerate the decay rate of the genes of the elite antibodies. It works on the condition that the affinity of the locally optimal antibody continues to increase. That is, under the condition of ensuring the classification performance of the optimal feature subset, it is accelerated to search in the direction of a smaller number of features. The calculation formula of the acceleration factor is shown in the following formula:

```
Function GeneratePop
  Initialize population P normal Cauchy distribution of Formula (3)
  For each Ab in P do
    The gene locus is transformed according to the threshold value η
  End for loop
  Return P
End function
```

Algorithm 1: Pseudo-code of population initialization

$$\delta = 0.5 - \frac{t}{T_{\max}}. \tag{5}$$

Among them, t is the current number of iterations, and $T_{\max}$ is the total number of iterations.

As the number of iterations increases, the value of δ will gradually tend to −0.5 from 0.5. In each iteration process, it is first necessary to determine the mutation loci of each antibody in mutation set C. The mutation loci are jointly determined by the generated Cauchy random number sequence and the transformation threshold value δ. Therefore, the threshold δ has an important effect on the position and quantity of the mutated loci of an antibody. It can be seen from the schematic diagram of the probability density function of the Cauchy distribution in Figure 1. When the value of the conversion threshold is smaller, there will be more "1" loci in an antibody, that is, the more loci involved in mutation, and vice versa. Therefore, when the conditional lethal mutation is used, the algorithm will speed up the search in the direction of fewer features. When the fitness value change of the local optimal antibody meets the mutation conditions, Figure 2 describes the genetic changes when an antibody performs lethal mutation operation.

Algorithm 2 lists the main steps to perform a lethal mutation operation on the elite antibodies selected from the population in each iteration.

In the classical clonal selection algorithm, the update of the population is carried out on the premise of maintaining the population number unchanged. In the HFIA algorithm, since the using of a lethal mutation strategy, the number of genes in the antibody decays rapidly during the mutation process. The purpose is to guide the algorithm to search in the direction of fewer features, so as to obtain high-quality feature subsets with fewer features. Compared with other intelligent algorithms based on metaheuristics, the search of the algorithm is not completely random. The advantage of this strategy is that the algorithm's search is better guided. On the other hand, this also helps to reduce the complexity and the computational cost of the algorithm itself. But correspondingly, it is also easier to cause the algorithm to fall into a locally optimal solution. In order to eliminate the risk of falling into a local optimum due to excessively rapid fitness decay, it is necessary to enhance the diversity of the population during the iterative process.

Therefore, two strategies are adopted to compensate. On the one hand, the size of the population is expanded when the population is updated. To this end, the strategy of incremental update is adopted. That is, the number of updates for HFIA is N, while the number of updates in the classical algorithm is d ($d < N$). This does not mean that the population will continue to increase during iterations. Selection is adopted to keep the overall size of the population constant. On the other hand, a linear incremental regulator θ is added in the population update phase. Its purpose is to dynamically adjust the number of antibody genes that is newly added to the population according to iterative changes. That is, the mutation probability of individuals in the population is enhanced, so as to achieve the purpose of improving the diversity of the population. The calculation formula of the adjustment factor is shown in the following formula:

$$\theta = \frac{t}{T_{\max}}. \tag{6}$$

Among them, t is the current number of iterations, and $T_{\max}$ is the total number of iterations. For its specific implementation and application, please refer to the algorithm framework code part in next section.

3.5. The Proposed Algorithm Framework and Notation. In theory, the higher the dimension of the data, the more detailed the description of things. This plays an important role in some fields of research. But for classification problems, too much redundant feature data will cause a serious decline in the performance of the classifier and even lead to the problem of dimensional disaster. For the feature selection problem in high-dimensional data space, the main idea is to use a hybrid feature selection method. But how to combine different algorithms more effectively is worthy of further study by scholars. Through many experiments, this paper finds a more efficient hybrid feature selection method than the current literature reports to solve the feature selection problem of high-dimensional data. The method solves the problem of selecting optimal feature subsets for high-dimensional data through a two-stage screening operation. The HFIA algorithm framework is shown in Figure 3.

The algorithm evaluates and ranks all features in the data space by the Fisher scoring function in the first stage. The Fisher score algorithm calculates the mean and variance of the distances between different categories of features and within the same category. It identifies the importance of features through the calculated mean and variance. It is an effective filtering feature selection method and has the advantage of fast calculation speed. The calculation method will be briefly introduced in [42].

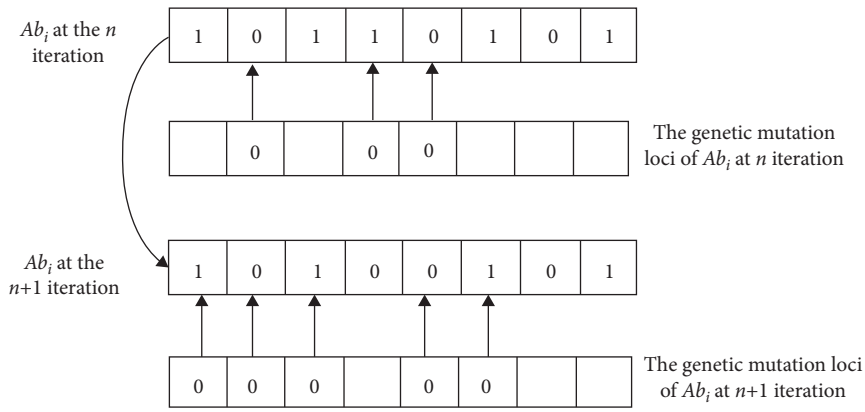


Figure 2: The schematic diagram of conditional lethal mutation of an antibody.

Function *Mutation*
$C^* \leftarrow$ CloneCopy (C, n), These n antibodies are cloned proportionally according to the fitness value
flag $\leftarrow$ Check(bestAb, k), Check the k consecutive fitness changes of bestAb, $k = 3$
If flag is true **then**
$\delta \leftarrow$ Calculate the value of the acceleration factor by formula (5)
End if
$C' \leftarrow$ Generate the genetic mutation loci according to the C^* and the threshold value as δ
$C \leftarrow C^*\{C'\} = 0$, Perform genetic lethal mutation
Return C
End function

Algorithm 2: Pseudo-code of *Mutation* based on gene lethal mutation mechanism

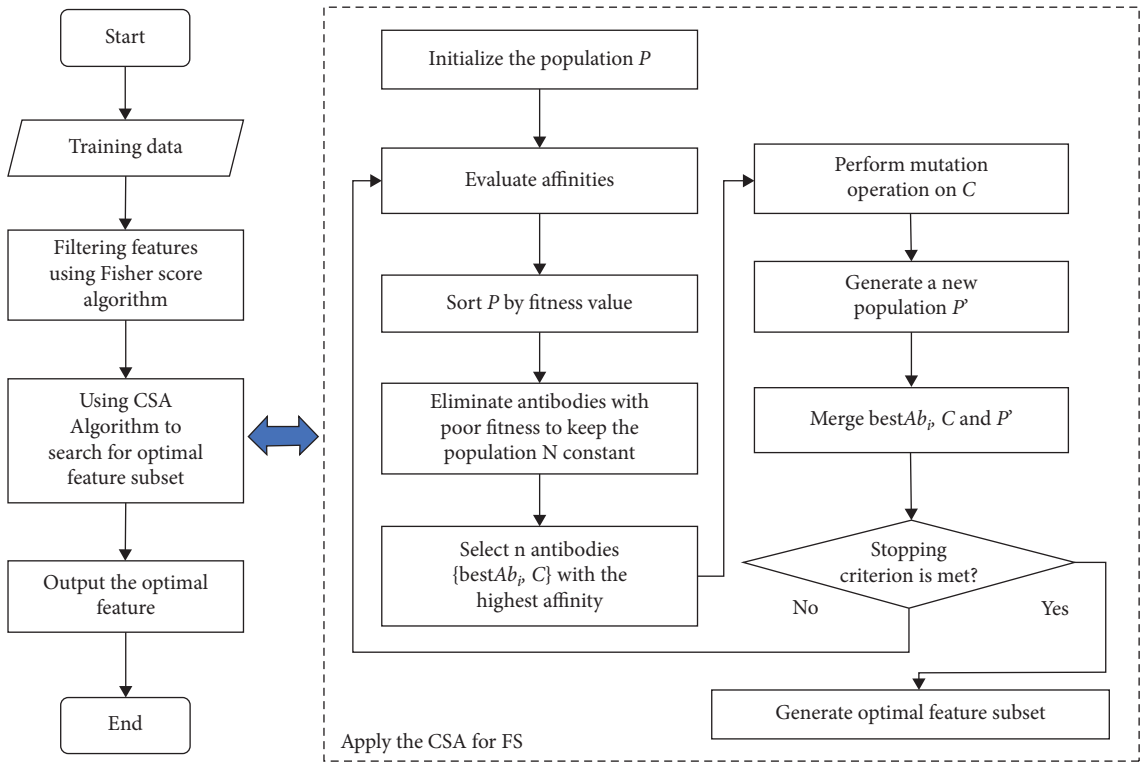


Figure 3: Framework of HFIA.

Given a set of labelled data samples, $\{Ab_i, Ab_j\}$, $Ab_j \in \{1, \ldots, c\}, i = 1, \ldots n$, where c is the number of categories, and n_k represents the number of data samples in the kth category. u_k^i represents the mean of all data samples on the ith feature. μ and σ are the mean and variance of the category k corresponding to the ith feature, respectively. The Fisher score of the ith feature can be calculated by the following formula:

$$F_k = \frac{\sum_{k=1}^{c} n_k \left(u_k^i - u^i\right)^2}{\sum_{k=1}^{c} n_k \left(\sigma_k^i\right)^2}, \tag{7}$$

where $\sum_{k=1}^{c} n_k (u_k^i - u^i)^2$ is the variance of the ith feature between different categories, and $\sum_{k=1}^{c} n_k (\sigma_k^i)^2$ is the variance of the kth feature within the same category.

After the classification importance scores of the features are obtained, the features can be filtered to obtain a reduced subset of candidate data. Compared with the full feature set, the feature subset obtained through the first stage has been greatly reduced in the number of features. Theoretically, for any filtering-type feature selection method, as long as an optimal threshold value is selected, the desired feature subset can be obtained. Although the selection of this optimal threshold value can be achieved by a simple exhaustive method, there is no guarantee that the feature combined with a high score is the one with the best quality. Because the feature score obtained by any univariate evaluation rule does not guarantee that the combination with a higher score is the optimal feature subset. In addition, experimental verification was performed to address this issue. On different datasets, the feature subsets are screened and classified by increasing the threshold. The experimental results show that the classification accuracy of the features scored and sorted by the Fisher score always oscillates within a certain range after being screened by different thresholds. Moreover, the peak value of its oscillation does not have a linear proportional relationship with the selected threshold value. The following Figure 4 is the relationship between the increase of the threshold value of the Fisher score of GLI-85 and the classification accuracy. The experimental results on other datasets are similar to this figure. It will not be repeated here.

Therefore, this paper adopts the artificial immune algorithm with a good global search performance to perform a secondary search on the feature subset after the initial screening. A hybrid feature selection method based on the Fisher filtering method combined with the wrapper method optimized by the artificial immune algorithm is constructed. After experimental analysis, considering both the quality of the optimal feature subset and the computational cost of the algorithm itself, this paper chooses the filter threshold value of the Fisher score to be 200. The structural framework and main steps of the HFIA algorithm are shown in the following algorithm 3.

In addition, Table 1 describes the important identifiers used in the algorithm.

4. Experiment Methodology

In this section, datasets used in the experiment are first introduced, then the performance evaluation criteria of the classification test are explained, and finally, the parameters setting of the HFIA algorithm in the experiment are described.

4.1. Datasets. In the experiments in this paper, a total of 25 real datasets are used to verify the performance of the proposed feature selection algorithm. These datasets cover varying numbers of features from 2000 to 22283. They are datasets from UCI Repository [43], feature selection datasets from Arizona State University [44], microarray datasets [45], and gene expression datasets [23], respectively. The UCI dataset is used in the evaluation of feature selection algorithms in many pieces of literature. In addition, the ASU feature selection dataset, microarray dataset, and gene expression dataset are specially selected to examine the performance of the algorithm on high-dimensional datasets. Table 2 shows the details of these datasets.

4.2. Performance Evaluation Criteria. In this paper, the cross-validation [46] is used to evaluate the accuracy of the classification algorithm. It is a commonly used validation technique and is widely used to evaluate the performance of machine learning models. The average classification accuracy of KNN is used to evaluate the quality of the selected optimal feature subset in this paper. The classification accuracy, the number of features of the optimal feature subset, and the average and deviation of the computational cost obtained from the experimental results are all statistical results after the algorithm runs 20 times independently on each dataset. And based on these statistical results, the performance of the algorithm is evaluated. For other parameters in the comparative experiment, the setting values described in the corresponding literature are used.

4.3. Parameter Settings. All experiments are performed on a PC with an Intel Core i5 and 8 GB of RAM. Also, all algorithms are performed on different datasets using the same settings. In all experiments, the parameter configuration of the HFIA algorithm is as follows. The maximum number of consecutive iterations of the algorithm is $T = 50$, the population size $N = 10$, the select rate $c_r = 0.5$, and the initial transformation threshold value of Cauchy random numbers $\eta = -0.2$. The parameter w of the fitness function is set to 0.99. According to the experimental analysis of the Fisher algorithm in Section 3.5, this paper sets its filter threshold value to be 200.

5. Experiments and Discussion

In this section, the proposed feature selection method HFIA is comprehensively evaluated and analyzed through experiments. Firstly, the performance of HFIA on all 25 datasets involved in the experiment is analyzed. The experimental results are compared with the results using full features. These comparisons include the reduction degree of redundant features and the improvement of the classifier performance. Secondly, the HFIA algorithm is compared with various feature selection methods reported in other

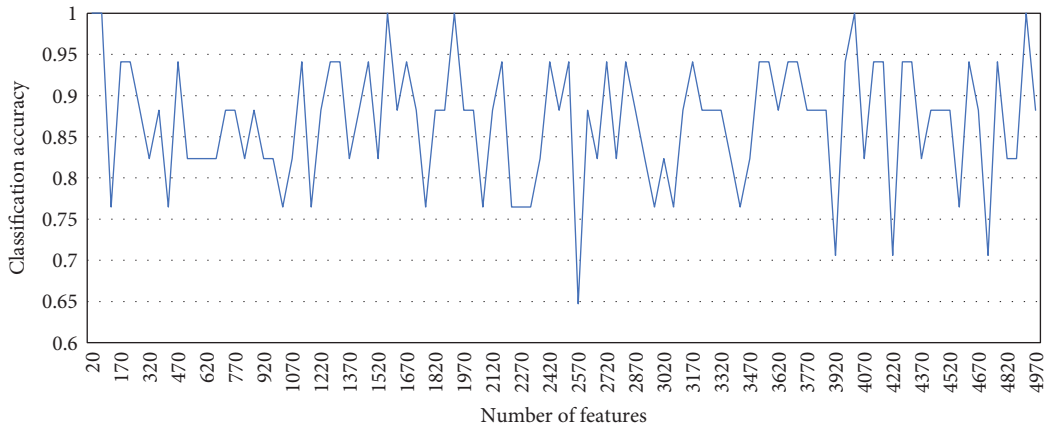


Figure 4: Relationship between the Fisher score screening threshold and classification accuracy of GLI-85.

Input:
Training sample trData
The antibody group size N
Maximum number of iterations $T_{\max}$
Select rate c_r. Proportion of antibodies selected for cloning
Output:
The optimal feature subset R_S.
/*Initialization*/, Set the initial parameters of the algorithm
Apply filtering algorithm (Fisher score) and generate feature subsets *ftD*
P ← GeneratePop (N, *ftD*), Generate initial population P based on feature subset *ftD* and Algorithm (1)
For $i \le T_{\max}$ **Do**
fitness ← FitnessFun (P), Evaluate affinities for antibodies in population P by formula (1)
sort(P), Sort P by fitness value in descending order
Select (P, N), Eliminate antibodies with poor fitness and keep the population N constant
{bestAb_i, C} ← Select (P, c_r), Select n antibodies with the highest affinity
Mutation(C), Submit C to Algorithm (2) for mutation operation
P' ← Generate a new population based on Algorithm (1) and threshold value θ by formula (6)
$M \leftarrow \{\text{best}Ab_i \cup C \cup P'\}$
$P \leftarrow M$, Update P by M
End for loop
$R_S = best(P)$
Return R_S

Algorithm 3: HFIA for feature selection.

Table 1: The description of symbols.

Symbols	Description
trData	The training sample data
ftD	Feature subset by the Fisher score
P	The set of antibodies
bestAb	The local optimal antibody
R_S	Optimal feature subset returned
N	The population size
n	$n = c_r \times N$, the selection pool size
fitness	The fitness of antibodies in population P
δ	The adaptive linear acceleration factor
θ	The linear increment factor
c_r	Select rate
$T_{\max}$	The maximum number of iterations
C	The antibody selection set
P'	The antibody mutation set
M	Population update set
η	Cauchy transform threshold for initial population

literature. These feature selection methods include several classical univariate filtering feature selection algorithms and a variety of hybrid feature selection methods reported in the latest literature. These analyses and comparisons include the following three aspects. They are the classification quality and the number of features of the optimal feature subset obtained, as well as the computational cost of the algorithm. In all tabular data, the best results of each standard are identified in bold.

5.1. Performance Evaluation. In this section, the effectiveness of the HFIA feature selection method in improving classifier performance is verified by experiments. Table 3 shows the quantitative comparison between the optimal feature subset (avgNfs) obtained using HFIA and the full features of the dataset. Figure 5 depicts a comparison

Table 2: The summary of the experimental datasets.

Id	Dataset	Feats	Ins	Cls	Id	Dataset	Feats	Ins	Cls
1	11Tumor	12533	174	11	14	ALL-AML-3C	7129	72	3
2	9Tumor	5726	60	9	15	ALL-AML-4C	7129	72	4
3	Brain Tumor1	5920	90	5	16	Lymphoma	4026	96	9
4	Brain Tumor2	10367	50	4	17	MLL	12582	72	3
5	DLBCL	5469	77	2	18	Ovarian	15154	253	2
6	Leukemia1	5327	72	3	19	SRBCT	2308	83	4
7	Leukemia2	7129	72	4	20	GLI-85	22283	85	2
8	Lung	12600	203	5	21	NCI9	9712	60	9
9	Prostate tumor	10509	102	2	22	SMK-CAN-187	19993	187	2
10	Leukemia3	11225	72	3	23	TOX-171	5748	171	4
11	Colon	2000	62	2	24	Orlraws10P	10304	100	10
12	CNS	7129	60	2	25	Pixraw10P	10000	100	10
13	ALL-AML	7129	72	2					

Note. Feats, Ins, and Cls represent features, instances, and classes, respectively.

Table 3: Quantitative comparison between the optimal feature subset of HFIA and full features.

Id	Dataset	Features	avgNfs/std	ID	Dataset	Features	avgNfs/std
1	Leukemia1	5327	1.8/0.41	14	ALL-AML-3C	7129	1.8/0.41
2	DLBCL	5469	2.15/2.11	15	ALL-AML-4C	7129	3.6/1.9
3	9Tumor	5726	17.95/4.64	16	Lymphoma	4026	6.95/3.34
4	Brain Tumor1	5920	12.75/3.12	17	MLL	12582	1.8/0.41
5	Prostate tumor	10509	2.5/1.15	18	Ovarian	15154	1.65/0.49
6	Leukemia2	7129	7.4/5.34	19	SRBCT	2308	2.8/0.7
7	Brain Tumor2	10367	3.6/1.9	20	GLI-85	22283	2/0.67
8	Leukemia3	11225	1.95/0.6	21	NCI9	9712	15/6.19
9	11Tumor	12533	24.35/3.1	22	SMK-CAN-187	19993	6.2/2.94
10	Lung	12600	7.35/4.85	23	TOX-171	5748	19.6/7.87
11	Colon	2000	1.7/0.82	24	Orlraws10P	10304	3.7/0.95
12	CNS	7129	1/0	25	Pixraw10P	10000	2/0
13	ALL-AML	7129	1.1/0.32				

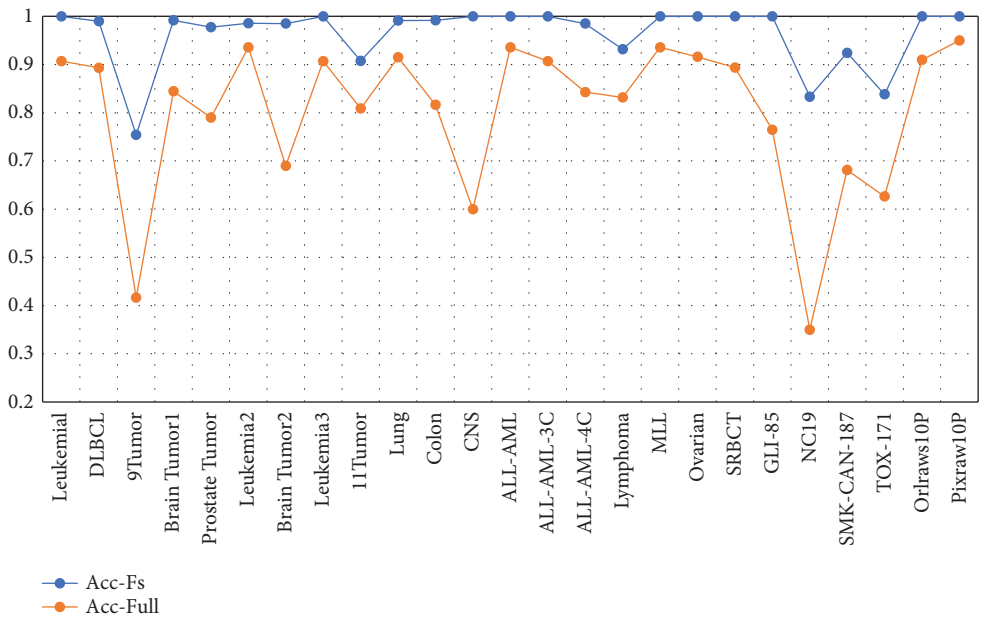


Figure 5: Comparison of KNN accuracy between HFIA and full.

between the classification accuracy obtained by the KNN classifier with or without HFIA feature selection. It should be noted that the results in Table 3 and Figure 5 are the average number and average classification accuracy of the optimal feature subsets of each dataset after repeated execution 20 times.

As can be seen from Table 3, HFIA achieves a very good performance on all experimental datasets in terms of removing feature redundancy. The removal rate of feature redundancy for all datasets is above 99%. According to statistics, in all datasets participating in the experiment, the number of features of the optimal feature subset screened by HFIA is within 0.34% of the total number of features. Among them, *TOX-171* has the largest proportion of the average number of features in the optimal feature subset, with a ratio of 0.34%. *GLI-85* has the smallest proportion of the number of features in the average optimal feature subset, and its ratio is only 0.00898%.

From the perspective of improving the classification performance of the classification algorithm, in all datasets, the HFIA method improves the classifier performance by 5%–48.333%. On 52% of the datasets, HFIA improves classification performance by more than 10%. On 28% of the datasets, it improves classification performance by more than 20%. On 12% of the datasets, it improves classification performance by more than 30%. The datasets with the highest classification performance improvement are *NCI9* and *CNS*. On these datasets, the performance of the classifier is improved by more than 40%.

From the comparative analysis of the above two aspects, it can be concluded that HFIA achieves better classification accuracy than the entire feature space with a very small number of features. This fully demonstrates the effectiveness of HFIA in eliminating redundant features.

5.2. Comparative Analysis. In order to verify the advanced performance of the proposed algorithm, this paper compares and analyses HFIA with several feature selection algorithms reported in other literature. These feature selection methods include 5 classical feature selection algorithms and 14 hybrid feature selection methods reported in the latest literature. These comparative analyses include the following three aspects. They are the number of features and classification accuracy of the obtained optimal feature subset and the computational cost paid by the algorithm, respectively. In all tabular data, the best result for each criterion is identified in bold. It should be noted that the experimental data of the comparison algorithm are all from the corresponding literature, and our algorithm adopts the same settings as the comparative literature.

5.3. Comparison with Classical Feature Selection Methods. This paper conducts comparative experiments with 5 classical feature selection methods on 10 benchmark datasets. The five methods are as follows: CFS (statistical-based) [47], FCBF (information theoretical-based) [48], ReliefF (similarity-based) [49], SBMLR (sparsity-based) [50], and SPEC (graph theory-based) [51]. The experimental results are shown in Tables 4–6. Table 4 presents a comparison of the classification accuracy of the optimal feature subsets obtained by different feature selection methods. Table 5 describes the comparison of the number of feature subsets for different algorithms to achieve optimal accuracy. Table 6 describes the computational cost of all algorithms to achieve optimal accuracy on these datasets. The classification accuracy data in the table is the best value obtained after 20 runs on each dataset. In this comparative experiment, all the results are obtained with the same classification algorithm and experimental parameter settings. It should be noted that the experimental data of the five classical feature selection methods in the table are all from the literature [23].

In terms of improving the performance of the classifier, the following results can be obtained from the observation and comparison of the data in Table 4. In the 10 datasets participating in the experiment, the classification accuracy of HFIA on all datasets is higher than that of the other 5 classical feature selection methods. According to the statistics in Table 3, on these datasets, the classification accuracy obtained by HFIA is 4.11%–32% higher than the maximum value of the other five algorithms. On 40% of the dataset, HFIA outperforms the maximum classification accuracy obtained by other methods by more than 10%. The highest proportion of classification accuracy is *9Tumor* and *Brain Tumor2*. On these datasets, the performance gains of the classifiers are more than 30% higher than the maximum value of other methods. This fully shows that compared with these five classical feature selection methods, the HFIA method is the best in improving the performance of the classifier.

In terms of reducing redundant features, the following results can be drawn from the data in Table 5. In these 10 datasets, the optimal feature subset obtained by HFIA has a lower number of features than other methods. It is only 2.94%–23.08% of the minimum value of other methods. On 80% of the datasets, the optimal subset obtained by HFIA has less than 15% of the minimum features of other methods. On 60% of the datasets, the number of features is below 10% of the minimum of other methods. The smallest proportion of features is *DLBCL*, *Prostate Tumor*, *Leukemia3*, and *Lung*. On these datasets, the number of features of the optimal feature subset obtained by HFIA is all below 7% of the minimum value of other methods. This fully shows that the HFIA method has the best effect in eliminating redundant features compared with these five classical feature selection methods.

In terms of the computational cost of the algorithm, the following results can be drawn from the data in Table 6. Among the five classical feature selection methods involved in the experiment, the SPEC method has the lowest computational cost. It is the fastest on all datasets. This is followed by ReliefF and FCBF methods, which are close in the computational cost on 70% of the dataset and outperform the rest of the methods. Again, the SBMLR method, which outperforms the CFS method on all datasets. Undoubtedly, the CFS method is the most computationally expensive among these 5 classical feature selection methods. By

Table 4: The comparison of the best classification accuracy.

Dataset	CFS	FCBF	ReliefF	SBMLR	SPEC	HFIA
Leukemia1	95.89	91.61	94.46	93.04	94.29	**100**
DLBCL	91.96	93.21	92.82	92.75	74.11	**100**
9Tumor	56.67	41.67	61.67	48.33	38.33	**91.667**
Brain Tumor1	86.67	80	85.78	82.22	84.44	**100**
Prostate tumor	94.09	91.12	92.09	95.09	85.36	**100**
Leukemia2	88.57	85.89	91.29	84.64	90	**100**
Brain Tumor2	68	64	68	64	48	**100**
Leukemia3	94.01	93.14	94.29	93.04	84.29	**100**
11Tumor	83.91	82.94	84.91	70.13	83.3	**97.059**
Lung	93.31	92.06	90.17	92.62	81.29	**100**

Table 5: Comparison of the number of optimal feature subsets obtained by different methods.

Dataset	CFS	FCBF	ReliefF	SBMLR	SPEC	HFIA
Leukemia1	97	49	603	14	2489	**1**
DLBCL	88	66	343	34	2371	**1**
9Tumor	47	32	544	26	1049	**6**
Brain Tumor1	142	106	771	27	5388	**4**
Prostate tumor	59	49	433	32	405	**1**
Leukemia2	119	71	645	18	3544	**2**
Brain Tumor2	117	75	1171	21	4420	**2**
Leukemia3	138	80	978	17	4917	**1**
11Tumor	379	394	1114	15	6158	**13**
Lung	550	453	1440	30	2378	**2**

Table 6: Computational cost comparison of different feature selection methods.

Dataset	CFS	FCBF	ReliefF	SBMLR	SPEC	HFIA
Leukemia1	741.57	1.49	1.3	0.92	0.19	4.8145
DLBCL	685.15	1.52	1.37	2.01	0.34	4.194
9Tumor	652.7	1.53	1.11	7.14	0.41	5.4977
Brain Tumor1	1248.17	2.69	2.04	9.2	0.66	5.8385
Prostate tumor	572.82	3.57	2.21	9.84	0.83	4.9926
Leukemia2	810.32	1.68	1.63	10.79	0.99	5.255
Brain Tumor2	765.5	2.84	1.54	11.5	1.16	5.3947
Leukemia3	677.51	4.04	2.84	13.46	1.4	5.341
11Tumor	4681.4	25.06	15.54	13.87	2.28	7.0041
Lung	10029	37.95	16.69	28.13	3.11	6.2198

comparing the data in the table, it can be concluded that the HFIA method proposed in this paper outperforms the SBMLR method on 80% of the datasets. It outperforms FCBF and ReliefF methods on 20% of the dataset. It outperforms the CFS method on all datasets. It can be concluded that the computational cost of HFIA is between SBMLR and FCBF. This fully demonstrates that HFIA is very competitive in terms of computational cost control, even compared with classical filtering-type feature selection methods known for their speed.

After summarizing the above analysis, the following conclusions can be drawn. In terms of computational cost alone, the HFIA algorithm proposed in this paper is comparable to the classical feature selection method known for its speed. Moreover, the obtained feature subset is much better than these 5 classical feature selection methods in terms of quantity and performance improvement of the classifier. To sum up, compared with the five classical feature selection methods, the HFIA method proposed in this paper can obtain a higher-quality feature subset while taking into account the computational cost.

5.4. Comparison with Other Hybrid Feature Selection Methods. This paper conducts comparative experiments on 25 benchmark datasets with 14 other feature selection methods for high-dimensional data reported in the latest literature. Tables 7–9 describe the comparison of experimental results between HFIA and the feature selection method mentioned in [23]. In the literature, the authors propose an evolutionary multitask-based feature selection method (PSO-EMT) and use it to solve the classification

Table 7: Comparison of classification accuracy of optimal feature subsets obtained by different methods.

Dataset	PSO	CSO	AMSO	VLPSO	PSO-EMT	HFIA
Leukemia1	80.60/2.55	90.79/2.88	94.01/1.58	93.31/2.34	91.11/2.79	**100/0**
DLBCL	83.67/1.52	94.60/3.26	94.10/1.95	86.51/2.88	93.76/2.80	**99/2.44**
9Tumor	42.72/1.42	59.78/3.55	50.11/3.61	54.94/4.80	58.00/4.02	**75.42/7.39**
Brain Tumor1	73.73/2.21	80.41/3.93	72.67/3.79	71.19/3.52	87.37/1.50	**99.17/2.04**
Prostate tumor	84.50/1.64	79.95/3.18	89.58/1.35	88.74/2.23	89.65/1.82	**97.75/2.55**
Leukemia2	78.61/2.02	80.83/2.28	87.52/2.00	85.82/2.96	90.07/2.47	**98.57/3.01**
Brain Tumor2	61.99/2.91	80.73/5.62	74.96/3.48	66.78/4.10	72.27/4.09	**98.5/3.66**
Leukemia3	89.83/1.00	91.49/3.84	94.45/1.04	91.56/1.67	94.51/1.50	**100/0**
11Tumor	71.81/1.75	83.50/1.70	83.10/1.31	80.92/2.39	86.15/1.45	**90.74/4.19**
Lung	78.77/1.53	88.94/1.75	89.97/1.80	89.55/1.68	91.09/0.94	**99.13/1.6**

Table 8: Comparison of the number of optimal feature subsets obtained by different methods.

Dataset	PSO	CSO	AMSO	VLPSO	PSO-EMT	HFIA
Leukemia1	2615.5	170.12	51.49	54.7	198.4	**1.8/0.41**
DLBCL	2681	30.08	50.56	48.14	83.55	**2.15/2.11**
9Tumor	2811.9	220.34	52.16	47.05	263.09	**17.95/4.637**
Brain Tumor1	2917.2	207.61	93.54	26.83	351.21	**12.75/3.117**
Prostate tumor	2926.6	207.98	44.36	35.97	149.86	**2.5/1.147**
Leukemia2	3513.8	389.4	71.54	53.39	224.44	**7.4/5.336**
Brain Tumor2	5117.2	90.43	62.08	81.46	499.69	**3.6/1.903**
Leukemia3	5535.7	88.64	57.19	35.23	268.08	**1.95/0.605**
11Tumor	6205	589.36	319	249.3	541.45	**24.35/3.104**
Lung	6234.7	230.41	193.47	176	617.61	**7.35/4.848**

Table 9: Computational cost comparison of different feature selection methods.

Dataset	PSO	CSO	AMSO	VLPSO	PSO-EMT	HFIA
Leukemia1	41.2	247.29	6.8	6.09	9.28	**5.3494/0.252**
DLBCL	47.59	389.67	8.34	7.18	7.02	**4.7939/0.34**
9Tumor	39.18	370.4	**5.52**	5.65	8.09	6.4389/0.396
Brain Tumor1	66.65	457.24	11.65	9.55	15.43	**6.4799/0.303**
Prostate tumor	78.77	410.83	14.31	11.05	16.88	**5.5481/0.441**
Leukemia2	66.09	445.99	9.66	8.96	12.19	**6.0048/0.485**
Brain Tumor2	80.5	945.7	12.06	11.76	11.51	**5.7796/0.334**
Leukemia3	120.64	1837.91	15.64	15.94	14.72	**5.7177/0.223**
11Tumor	418.54	6278.54	91.22	67.41	106.53	**7.5514/0.651**
Lung	574.17	5419.71	255.32	78	134.59	**6.7417/0.313**

problem of high-dimensional data. It performs well in improving classification accuracy and computational cost. This paper compares this method with 4 other feature selection algorithms on 10 gene expression datasets. These datasets all have high dimensionality, the number of features varies from 5327 to 12600, and the number of samples is small. These 4 feature selection methods are PSO, CSO, AMSO, and VLPSO. In the experiment, the classification results of the cross-validation of the KNN algorithm are used as the basis for the performance evaluation of the algorithm. The experimental results are shown in Tables 7–9. Table 7 describes the comparison of the average classification accuracy between the HFIA algorithm and the five methods after 20 repetitions on all experimental datasets. Table 8 describes the comparison between the average numbers of optimal feature subsets obtained by different methods. Table 9 describes the average computational cost of all algorithms on these 10 datasets.

In terms of improving the performance of the classifier, the following results are obtained by observing and comparing the data in Table 7. Among the 10 datasets participating in the experiment, the classification accuracy of HFIA is higher than that of the other 5 feature selection methods on all datasets. After statistical analysis of the data in the table, the following results are obtained. On these datasets, the classification accuracy obtained by HFIA is 4.4%–17.77% higher than the maximum value of the other five algorithms. On 80% of the datasets, HFIA outperforms the maximum classification accuracy obtained by other methods by more than 5%. On 30% of the datasets, HFIA outperforms the maximum values of other methods by more than 10%. The highest proportion of classification accuracy is *9Tumor* and

Brain Tumor2. On these datasets, the classification accuracy obtained by HFIA is more than 17% higher than the maximum value of other methods. This fully shows that the HFIA method is the best in improving the performance of the classifier compared with these five feature selection methods.

In terms of reducing redundant features, the following results can be drawn from the data in Table 8. In all datasets participating in the experiment, the number of features of the optimal feature subset obtained by the HFIA method is less than 50% of the minimum value of other methods. It is only 3.5%–47.52% of the minimum value of other methods. On 80% of the datasets, the number of features of the optimal feature subset obtained by the HFIA method is below 14% of the minimum value of the other methods. On 70% of the dataset, it is below 10% of the minimum of other methods. The optimal feature subset with the smallest proportion of features is *Leukemia1*, *Brain Tumor2*, *Leukemia3*, and *Lung*. On these datasets, the number of features of the optimal feature subset obtained by HFIA is less than 6% of the minimum value of other methods. This fully shows that the HFIA method has the best performance in eliminating redundant features compared with these five methods.

In terms of the computational cost of the algorithm, the following results can be drawn from the data in Table 9. Among the five feature selection methods participating in the comparative experiments, the VLPSO method has the lowest computational cost. Its computational cost is lower than the other 4 methods on 70% of the datasets. This is followed by AMSO and PSO-EMT methods, which are close in the computational cost on 60% of the dataset and outperform the rest of the methods. Next is PSO, which outperforms CSO methods on all datasets. Undoubtedly, the CSO method is the most computationally expensive of them all. By comparing the data in the table, it can be concluded that the HFIA method proposed in this paper outperforms the five methods on 90% of the datasets and only 8.6%–87.84% of the minimum value of other methods. The computational cost on 80% of the datasets is below 70% of the lowest value of other methods. The computational cost on 30% of the datasets is below 40% of the lowest value of other methods. Among them, *11Tumor* and *Lung* have the lowest computational cost, which is less than 12% of the lowest value of other methods. This fully shows that, compared with these five feature selection methods, HFIA has significant advantages in controlling the computational cost.

Through the above comparative analysis, we can draw the following summary. The optimal feature subset obtained by HFIA is better than the other five methods in average classification accuracy and average number. Moreover, its computational cost on 90% of the datasets is better than these five methods. Therefore, the following conclusions can be further drawn. Compared with these five feature selection methods, the HFIA algorithm proposed in this paper has strong competitive advantages in both the quality of the optimal feature subset and the computing speed of the algorithm. This fully proves the progressiveness of the HFIA algorithm in solving the feature selection problem of high-dimensional data.

Tables 10–12 describe the comparison of experimental results between HFIA and the feature selection method mentioned in [45]. In the literature, the authors propose a hybrid feature selection method based on the binary Jaya algorithm (TOPSIS-Jaya) and use it to solve the classification problem of high-dimensional microarray data. This method performs well in improving classification accuracy and computational cost. This paper compares this method and four other advanced feature selection algorithms on 10 microarray datasets. The four feature selection methods are HSAMB [52], CFS-iBPSO [53], PSO-DT [54], and MBEGA [55], respectively. The experimental results are shown in Tables 10–12. Table 10 describes the comparison of the average classification accuracy of the HFIA algorithm with the 5 feature selection methods on these 10 datasets. Table 11 describes the comparison between the average numbers of optimal feature subsets obtained by these methods. Table 12 describes the average computational cost of all algorithms on these datasets.

In terms of improving the performance of the classifier, the following results are obtained by comparing the data in Table 10. Among the five feature selection methods involved in the comparative experiments, CFS-iBPSO obtained the highest classification accuracy, outperforming the other four methods on 70% of the datasets. The second is the TOPSIS-Jaya method, which outperforms the remaining 3 methods on 70% of the datasets. Next is the HSAMB method, which outperforms the remaining 2 methods on 80% of the datasets. Finally, there are PSO-DT and MBEGA, which perform relatively similarly on all datasets. The statistical results of the proposed HFIA method on all datasets are very similar to the CFS-iBPSO method. The average classification accuracy obtained on 70% of the datasets is greater than or equal to the best value of other methods. Moreover, its performance on 60% of the datasets achieved an average classification accuracy of 100%. This fully shows that the HFIA method has strong competitiveness in improving the performance of the classifier compared with these five feature selection methods.

In terms of the control of the number of features of the optimal feature subset, the following results can be obtained by comparing the data in Table 11. In all datasets participating in the experiment, the HFIA method outperformed the other five methods on 90% of the datasets, only 13.46%–74.24% of the minimum values of the other methods. Moreover, on 80% of the datasets, the number of features of the optimal feature subset obtained by the HFIA method is below 57% of the minimum value of other methods. On 50% of the dataset, it is below 32% of the minimum of other methods. The smallest proportion of features in the optimal feature subset is *CNS*, *ALL-AML*, and *MLL*. On these datasets, the number of optimal feature subsets obtained by HFIA is less than 28% of the minimum value of other methods. This fully shows that, compared with these five feature selection methods, the HFIA method is very advantageous in eliminating redundant features.

Table 10: Comparison of classification accuracy of optimal feature subsets obtained by different methods.

Dataset	TOPSIS-Jaya	HSAMB	CFS-iBPSO	PSO-DT	MBEGA	HFIA
Colon	97.76	90.27	94.89	90.32	86.66	**99.17/0.026**
CNS	96.22	84.17	95.84	58.33	72.21	**100/0**
ALL-AML	**100**	99.34	**100**	95.83	95.89	**100/0**
ALL-AML-3C	**100**	99.18	**100**	95.83	96.64	**100/0**
ALL-AML-4C	**99.72**	96.79	97.63	94.44	91.39	98.5/0.037
Lung	94.24	–	**100**	**100**	98.96	99.13/0.016
Lymphoma	98.33	99.99	**100**	98.5	97.68	93.16/0.027
MLL	99.62	99.55	**100**	94.04	94.33	**100/0**
Ovarian	99.52	99.81	**100**	97.23	99.71	**100/0**
SRBCT	**100**	99.57	**100**	92.49	99.23	**100/0**

Table 11: Comparison of the number of optimal feature subsets obtained by different methods.

Dataset	TOPSIS-Jaya	HSAMB	CFS-iBPSO	PSO-DT	MBEGA	HFIA
Colon	18.9	4.16	4.2	643.3	24.5	**1.7/0.823**
CNS	8.7	7.43	10.5	1486	20.5	**1/0**
ALL-AML	16.1	5	4.3	1468	15.8	**1.1/0.316**
ALL-AML-3C	6.6	5.84	6	1294.1	20.1	**1.8/0.41**
ALL-AML-4C	19.5	6.37	20.7	1845	26.2	**3.6/1.903**
Lung	9.9	—	10.6	1657	14.1	**7.35/4.848**
Lymphoma	15.2	**3.75**	24	1346	34.3	6.95/3.342
MLL	12.9	6.6	30.8	4847	32.1	**1.8/0.41**
Ovarian	18.5	5.73	3.3	3594.2	9	**1.65/0.489**
SRBCT	15.8	8.9	34.1	874	60.7	**2.8/0.696**

Table 12: Computational cost comparison of different feature selection methods.

Dataset	TOPSIS-Jaya	HSAMB	CFS-iBPSO	MBEGA	HFIA
Colon	12.71	142	39.27	70.6	**3.7616/0.563**
CNS	14.79	101	78.43	81.1	**8.9231/0.51**
ALL-AML	15.07	102	141.48	112.3	**3.8569/0.234**
ALL-AML-3C	16.4	233	204.15	176.6	**12.3797/0.789**
ALL-AML-4C	16.7	141	321.33	234.3	**5.7796/0.334**
Lung	35.03	—	311.22	1041.7	**6.7417/0.313**
Lymphoma	17.42	92	366.24	142.6	**14.0521/0.523**
MLL	18.33	152	245.71	182.1	**12.8591/1.083**
Ovarian	68.49	3000	92.9	2689.5	**12.2007/0.796**
SRBCT	16.38	188	302.81	246.2	**6.0331/0.259**

In terms of the computational cost of the algorithm, the following results can be obtained after comparing the data in Table 12. Among the six feature selection methods participating in the comparative experiments, the HFIA method outperforms other methods on all datasets. Moreover, its computational cost is only 17.8%–80.7% of the minimum value of other methods. Its computational cost on 60% of the datasets is below 37% of the lowest value of other methods. Its computational cost on 40% of the datasets is below 30% of the lowest value of other methods. Among them, Ovarian and Lung have the lowest computational cost, which is less than 20% of the lowest value of other methods. This fully shows that, compared with these five feature selection methods, the HFIA method has the lowest computational cost.

The following conclusions can be drawn from the above analysis. Compared with the five feature selection methods involved in the experiment, the HFIA algorithm has strong advantages in improving the classifier performance and reducing redundant features and computational cost. This shows that the HFIA algorithm can obtain higher quality feature subsets with less computational cost. This also fully proves the superiority of the HFIA algorithm proposed in this paper.

Tables 13 and 14 describe the comparison of experimental results between HFIA and the feature selection method mentioned in [56].

In the literature, the authors propose an evolutionary algorithm-based filter feature selection algorithm (TAGA) and use it to solve the classification problem of high-dimensional data. This method performs well in improving classification accuracy. This paper compares this method and 4 other algorithms on 8 benchmark datasets. The four feature selection methods are mRMR-mid [57], QPFS [58],

Table 13: Comparison of classification accuracy of optimal feature subsets obtained by different methods.

Algorithm	CLN	GLI	NCI	SMK	TOX	LYM	ORP	PIW
TAGA	94.03/0.8	98.8/0.0	79.5/0.4	75.2/0.5	77.7/0.8	94.3/0.5	99.8/0.4	97.0/0.0
mRMR-mid	98.4	95.3	65	68.4	72.5	**97.9**	97	96
QPFS	91.9	94.1	83.3	74.3	72.5	**97.9**	98	97
SPECCMI	93.5	96.5	80	71.1	77.2	93.8	94	95
CGA	95.2/1.3	95.9/0.6	78.8/2.4	72.7/1.5	74.3/2.6	94.1/1.0	98.5/1.0	97.2/0.4
HFIA	**99.17/2.64**	**100/0**	**83.33/5.56**	**92.43/3.32**	**83.82/3.73**	93.16/2.69	**100/0**	**100/0**

Note. CLN, GLI, NCI, SMK, TOX, LYM, ORP, and PIW in the table represent the datasets Colon, GLI-85, NCI9, SMK-CAN-187, TOX-171, Lymphoma, Orlraws10P, and Pixraw10P, respectively.

Table 14: Comparison of the number of optimal feature subsets obtained by different methods.

Algorithm	CLN	GLI	NCI	SMK	TOX	LYM	ORP	PIW
TAGA	10.6/8.1	14.8/4.3	40.5/3.5	13.1/4.4	23.6/6.5	20.3/3.7	13.4/4.7	8.1/0.7
mRMR-mid	**1**	5	43	**7**	28	22	20	7
QPFS	3	12	39	17	**15**	23	9	20
SPECCMI	32	16	38	13	24	36	21	11
CGA	7.9/3.9	16.2/7.0	39.0/5.8	8.8/6.3	28.8/3.9	30.3/7.0	13.6/3.6	10.2/5.1
HFIA	1.7/0.82	**2/0.67**	**15/11.19**	**6.2/2.94**	19.6/17.87	**6.95/3.34**	**3.7/0.95**	**2/0**

SPECCMI [59], and CGA [60], respectively. The experimental results are shown in Tables 13 and 14. Table 13 describes the comparison of the average classification accuracy of the HFIA algorithm with the five feature selection methods. Table 14 describes the comparison between the average numbers of optimal feature subsets obtained by these methods.

In terms of the classification accuracy of the selected optimal feature subset, the following results are obtained by comparing the data in Table 13. Among the six feature selection methods involved in the experiment, the HFIA method proposed in this paper outperforms the other five methods on 87.5% of the datasets. On these datasets, the classification accuracy obtained by HFIA is 0.03%–18.13% higher than the optimal values of the other five algorithms. Among them, the average classification accuracy obtained on 50% of the dataset is more than 3% higher than the best value of other methods. This fully shows that, compared with these five feature selection methods, the optimal feature subset obtained by HFIA has a strong competitive advantage in classification performance.

In terms of eliminating redundant features, the following results can be drawn from the comparison of the data in Table 14. In all datasets participating in the experiment, the HFIA method outperformed the other five methods on 75% of the datasets, only 28.57%–88.57% of the minimum values of the other methods, and on 62.5% of the datasets, the number of optimal feature subsets obtained by the HFIA method is below 42% of the minimum values of other methods. The smallest proportion of features in the optimal feature subset is *Lymphoma* and *Pixraw10P*. On these datasets, the optimal number of optimal feature subsets obtained by HFIA is less than 32% of the minimum value of other methods. This fully shows that, compared with these five feature selection methods, the redundancy of the optimal feature subset selected by the HFIA method is very advantageous.

The following conclusions can be drawn from the above analysis. Compared with the five feature selection methods involved in the experiment, the HFIA algorithm has strong advantages in improving the performance of the classifier and reducing redundant features. This shows that the optimal feature subset obtained by the HFIA algorithm has higher quality. This also fully proves the superiority of the HFIA algorithm proposed in this paper.

Table 15 describes the average computational cost of HFIA and the other 4 algorithms (TAGA, SFS, BE, and CGA) on these 8 datasets. After comparing the data in Table 14, the following results can be obtained. Among the five feature selection methods involved in the experiment, the HFIA method outperforms other methods on all datasets. Moreover, its computational cost is only 4.3%–23.83% of the minimum value of other methods. Its computational cost on 75% of the datasets is below 15% of the lowest value of other methods. The computational cost on 62.5% of the datasets is below 10% of the lowest value of other methods. Among them, *SMK-CAN*-187, *TOX*-171, and *Pixraw10P* have the lowest computational cost, which is less than 7.5% of the lowest value of other methods. This fully shows that, compared with these five feature selection methods, the HFIA method has the lowest computational cost.

Through the comparative analysis of the above experimental results, it can be concluded that, combined with the evaluation results of the two indicators of the quality of the optimal feature subset and the computational cost, the HFIA method has excellent competitive advantages in feature selection of high-dimensional data compared with the 14 feature selection methods reported in the latest literature.

5.5. Ablation Experiments. In order to verify the necessity of each part of the functional modules in the proposed model, ablation experiments are also performed. In this paper,

Table 15: Computational cost comparison of different feature selection methods.

Algorithm	CLN	GLI	NCI	SMK	TOX	LYM	ORP	PIW
TAGA	123	122	127	139	132	117	129	148
SFS	34	48	33	140	123	59	62	63
BE	183	223	162	649	587	294	288	301
CGA	158	181	168	183	174	161	199	183
HFIA	**3.76/0.56**	**4.46/0.2**	**5.79/0.06**	**9.73/0.88**	**5.29/0.19**	**14.05/0.52**	**5.69/0.27**	**4.66/0.06**

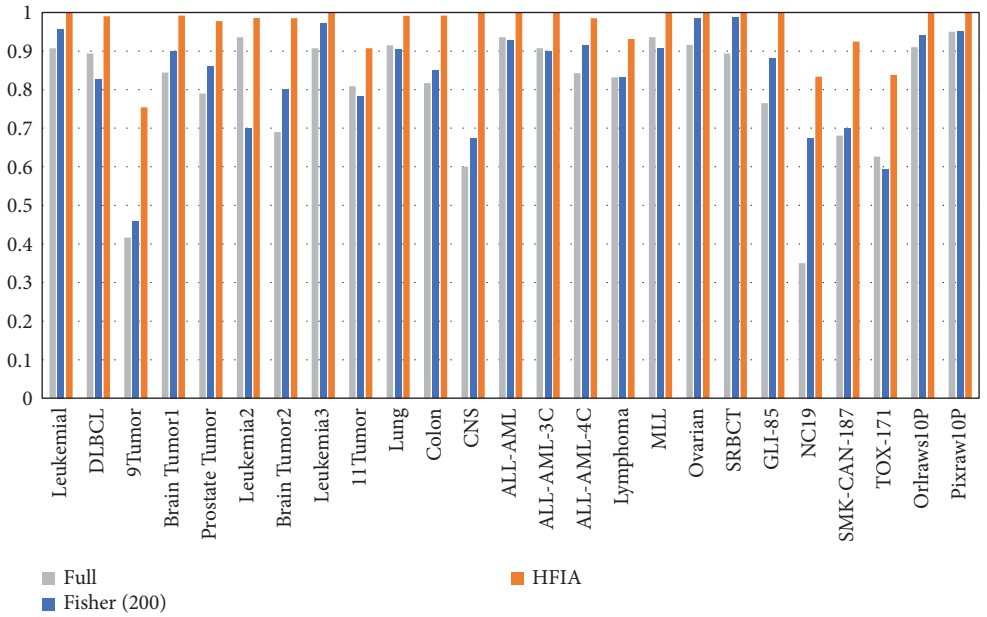


Figure 6: Ablation study results of the proposed method for the 25 datasets.

ablation tests are performed on all 25 datasets participating in the experiment. For the sake of simplicity, only the classification results of the KNN algorithm are used as the analysis indicators to conduct experiments. As shown in Figure 6, the experimental results under each metric are presented. Among them, "Full" represents the method of removing all feature subset evaluation modules. "Fisher (200)" represents a module that removes the search part of the artificial immune algorithm. HFIA means the fusion of all functional modules. Through the observation and analysis of the experimental results, the following results can be obtained. From the performance of each module on the test data set, the results obtained by HFIA are significantly higher than the other two schemes and have very obvious advantages. From the performance of the KNN algorithm in classification accuracy on all datasets, the fusion method of HFIA has a better performance advantage than any other individual method. Through the above analysis, the following conclusions can be drawn. For the model proposed in this paper, the fusion scheme of HFIA is effective, which is very helpful for the performance improvement of the classifier.

6. Conclusion

In this paper, an efficient hybrid feature selection method (HFIA) based on an artificial immune algorithm is proposed. The algorithm combines the Fisher filter algorithm and an improved artificial immune algorithm to optimize the search process of the optimal feature subset for high-dimensional data. According to the target requirements of feature selection, the method improves the population initialization and mutation strategy of the antibody in the algorithm, as well as the population update method.

In order to verify the effectiveness of the HFIA algorithm, we conducted many experimental verifications and analyses on 25 high-dimensional datasets with features ranging from 2000 to 22283. These experimental analyses cover the following three aspects. (1) The algorithm improves the classification performance of the classifier. We compared the experimental results obtained by HFIA with the results without feature selection. These analyses include the reduction of feature redundancy and the improvement of classification accuracy. (2) Comparative analysis with other feature selection methods. We compared the experimental results with the results of 19 feature selection methods mentioned in other pieces of literature. These feature selection methods include five classical feature selection algorithms and 14 hybrid feature selection methods reported in the latest literature. These comparative analyses include classification accuracy, the number of optimal feature subsets, and the computational cost of the algorithm. (3) The structural validity analysis of the algorithm itself. To verify the

necessity of each part of the functional modules in the proposed model, we conduct ablation experiments.

Based on the analysis results of the above three aspects, the following conclusions are drawn. The optimal feature subset obtained by the HFIA algorithm can improve the classification accuracy of the classifier to a great extent. Compared with the classical filtering feature selection method, the quality of the optimal feature subset obtained by the HFIA algorithm has great advantages, and its computational cost is also very competitive. Compared with the hybrid feature selection method proposed in the latest literature, the HFIA algorithm obtains the minimum number of selected feature subsets and better average classification accuracy at a lower computational cost. Therefore, it can fully illustrate the effectiveness and progressiveness of this method in solving the problem of feature selection of high-dimensional data.

In addition, it has to be said that although the HFIA algorithm has greatly improved the computational efficiency and the quality of the obtained feature subsets, there are still some problems to be solved. First, this paper uses the classification results of KNN as a criterion for evaluating the quality of candidate feature subsets. In order to obtain more accurate feature evaluation information, a fusion scheme of multiple metrics, such as rough set theory, can be considered in the evaluation of feature subsets in future research. Thirdly, because the selected optimal feature subsets are different in different classifiers, a fusion framework combining multiple feature selection algorithms and classification algorithms can be considered in order to obtain more effective results. In addition, how to obtain a better balance between classification accuracy, feature reduction rate, and computational cost is still a direction worthy of further research.

Data Availability

The datasets were taken from UCI Repository; Arizona State University; https://github.com/primekangkang/Genedata (microarray datasets); and https://ckzixf.github.io/dataset.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (No. 2020YFB1805400); in part by the National Natural Science Foundation of China (Nos. U1736212, U19A2068, 62032002, and 62002248); in part by the China Postdoctoral Science Foundation (Nos. 2019TQ0217, 2020M673277, and 2020M683345); in part by the Provincial Key Research and Development Program of Sichuan (No. 20ZDYF3145); in part by the Fundamental Research Funds for the Central Universities (No. YJ201933); and in part by the China International Postdoctoral Exchange Fellowship Program (Talent-Introduction).

References

[1] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm and Evolutionary Computation*, vol. 54, Article ID 100663, 2020.

[2] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognition*, vol. 42, no. 3, pp. 409–424, Mar 2009.

[3] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131–156, Jul. 1997.

[4] Z. Sun, *Parallel Feature Selection Based on MapReduce*, Springer, China, 2014.

[5] M. Rostami, K. Berahmand, E. Nasiri, S. Forouzandeh, and S. Forouzandeh, "Review of swarm intelligence-based feature selection methods," *Engineering Applications of Artificial Intelligence*, vol. 100, Article ID 104210, 2021.

[6] J. Kim, J. Kang, and M. Sohn, "Ensemble learning-based filter-centric hybrid feature selection framework for high-dimensional imbalanced data," *Knowledge-Based Systems*, vol. 220, Article ID 106901, 2021.

[7] S. N. Qasem and F. Saeed, "Hybrid feature selection and ensemble learning methods for gene selection and cancer classification," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, pp. 193–200, 2021.

[8] S. Manochandar and M. Punniyamoorthy, "Scaling feature selection method for enhancing the classification performance of Support Vector Machines in text mining," *Computers & Industrial Engineering*, vol. 124, pp. 139–156, Oct. 2018.

[9] S. Chidambaram and A. Sumathi, "Optimal feature selection for the classification of hyperspectral imagery using adaptive spectral–spatial clustering," *International Journal of Parallel Programming*, vol. 48, no. 5, pp. 813–832, Oct. 2020.

[10] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, pp. 1200–1205, 2015.

[11] J. G. Chen and H. Inbarani, "Hybrid Tolerance Rough Set–Firefly based supervised feature selection for MRI brain tumor image classification," *Applied Soft Computing*, vol. 46, pp. 639–651, Sep. 2016.

[12] V. Santucci, M. Baioletti, and A. Milani, "An algebraic framework for swarm and evolutionary algorithms in combinatorial optimization," *Swarm and Evolutionary Computation*, vol. 55, Article ID 100673, 2020.

[13] S. Zhang, C. K. M. Lee, H. K. Chan, K. L. Choy, and Z. Wu, "Swarm intelligence applied in green logistics: a literature review," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 154–169, Jan. 2015.

[14] S. Barak, J. H. Dahooie, and T. Tichý, "Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9221–9235, 2015.

[15] X. Y. Liu, Y. Liang, S. Wang, Z. Y. Yang, and H. S. Ye, "A hybrid genetic algorithm with wrapper-embedded approaches for feature selection," *IEEE Access*, vol. 6, pp. 22863–22874, 2018.

[16] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao, "A hybrid feature selection algorithm for gene expression data classification," *Neurocomputing*, vol. 256, pp. 56–62, 2017.

[17] W. Ma, X. Zhou, H. Zhu, L. Li, and L. Jiao, "A two-stage hybrid ant colony optimization for high-dimensional feature selection," *Pattern Recognition*, vol. 116, Article ID 107933, 2021.

[18] Z. Huang, C. Yang, X. Zhou, and T. Huang, "A hybrid feature selection method based on binary state transition algorithm and ReliefF," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, 1898 pages, 2019.

[19] C. Yan, J. Ma, H. Luo, and A. Patel, "Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets," *Chemometrics and Intelligent Laboratory Systems*, vol. 184, pp. 102–111, 2018.

[20] K. Hussain, N. Neggaz, W. Zhu, and E. H. Houssein, "An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection," *Expert Systems with Applications*, vol. 176, Article ID 114778, 2021.

[21] X. F. Song, Y. Zhang, D. W. Gong, and X. Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9573–9586, 2022.

[22] S. A. B. P. Xiao, C. S. R. Annavarapu, and S. Dara, "Clustering-based hybrid feature selection approach for high dimensional microarray data," *Chemometrics and Intelligent Laboratory Systems*, vol. 213, Article ID 104305, 2021.

[23] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An evolutionary multitasking-based feature selection method for high-dimensional classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7172–7186, 2022.

[24] J. Wan, H. Chen, Z. Yuan, T. Li, X. Yang, and B. Sang, "A novel hybrid feature selection method considering feature interaction in neighborhood rough set," *Knowledge-Based Systems*, vol. 227, Article ID 107167, 2021.

[25] X. Song, Y. Zhang, D. Gong, H. Liu, and W. Zhang, "Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data," *IEEE Transactions on Evolutionary Computation*, vol. 23, 2022.

[26] M. Burnet, *The Clonal Selection Theory of Acquired Immunity*, Vanderbilt University Press, Nashville, 1959.

[27] L. N. De Castro and F. J. Von Zuben, "The clonal selection algorithm with engineering applications," *Proceedings of GECCO*, vol. 8, pp. 36–39, 2000.

[28] B. Haktanirlar Ulutas and S. Kulturel-Konak, "A review of clonal selection algorithm and its applications," *Artificial Intelligence Review*, vol. 36, no. 2, pp. 117–138, 2011.

[29] R. Shang, L. Jiao, F. Liu, and W. Ma, "A novel immune clonal algorithm for MO problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, 2012.

[30] H. Dai, Y. Yang, H. Li, and C. Li, "Bi-direction quantum crossover-based clonal selection algorithm and its applications," *Expert Systems with Applications*, vol. 41, no. 16, pp. 7248–7258, 2014.

[31] N. Xu, Y. Ding, L. Ren, and K. Hao, "Degeneration recognizing clonal selection algorithm for multimodal optimization," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 848–861, Mar. 2018.

[32] X. Yan, P. Li, K. Tang, L. Gao, and L. Wang, "Clonal selection based intelligent parameter inversion algorithm for prestack seismic data," *Information Sciences*, vol. 517, pp. 86–99, 2020.

[33] W. Luo, X. Lin, T. Zhu, and P. Xu, "A clonal selection algorithm for dynamic multimodal function optimization," *Swarm and Evolutionary Computation*, vol. 50, Article ID 100459, 2018.

[34] M. Aladeemy, S. Tutun, and M. T. Khasawneh, "A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence," *Expert Systems with Applications*, vol. 88, pp. 118–131, 2017.

[35] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, 2016.

[36] M. M. Mafarja and S. Mirjalili, "Hybrid Whale Optimization Algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017.

[37] M. A. E. Aziz and A. E. Hassanien, "Modified cuckoo search algorithm with rough sets for feature selection," *Neural Computing & Applications*, vol. 29, no. 4, pp. 925–934, Feb. 2018.

[38] O. S. Qasim and Z. Y. Algamal, "Feature selection using particle swarm optimization-based logistic regression model," *Chemometrics and Intelligent Laboratory Systems*, vol. 182, pp. 41–46, 2018.

[39] K.-T. Lan and C.-H. Lan, "Notes on the distinction of Gaussian and Cauchy mutations," in *Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications*, pp. 272–277, Kaohsuing, Taiwan, November 2008.

[40] H. Wang, C. Li, Y. Liu, and S. Zeng, "A Hybrid Particle Swarm Algorithm with Cauchy Mutation," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, pp. 356–360, Honolulu, HI, USA, April 2007.

[41] W. c. Wang, L. Xu, K. w. Chau, and D. m. Xu, "Yin-Yang firefly algorithm based on dimensionally Cauchy mutation," *Expert Systems with Applications*, vol. 150, Article ID 113216, 2020.

[42] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," *Proc. 27th Conf. Uncertain. Artif. Intell. UAI*, pp. 266–273, 2011.

[43] D. Dua and C. Graff, *UCI Machine Learning Repository*, Beijing China, 2017.

[44] A. S. University, *Arizona State University's (ASU) Repository*, GitHub, San Francisco, California, U.S, 2018.

[45] A. Chaudhuri and T. P. Sahu, "A hybrid feature selection method based on Binary Jaya algorithm for micro-array data classification," *Computers & Electrical Engineering*, vol. 90, Article ID 10696, 2020.

[46] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, 2010.

[47] M. a. Hall and L. a. Smith, "Practical feature subset selection for machine learning," *Comput. Sci*, vol. 98, pp. 181–191, 1998.

[48] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution," *Proceedings, Twent. Int. Conf. Mach. Learn*, vol. 2, pp. 856–863, 2003.

[49] R. S. Marko and K. Igor, "Theoretical and empirical analysis of relief and rreliefF," *Machine Learning J*, vol. 53, pp. 23–69, 2003.

[50] G. Cawley, N. Talbot, and M. Girolami, "Sparse multinomial logistic regression via Bayesian L1 regularisation," *Advances in Neural Information Processing Systems*, vol. 19, 2006.

[51] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proceedings of the 24th International Conference on Machine Learning - ICML*, pp. 1151–1157, Oregon, Corvalis, USA, June 2007.

[52] S. S. Shreem, S. Abdullah, and M. Z. A. Nazri, "Hybridising harmony search with a Markov blanket for gene selection problems," *Information Sciences*, vol. 258, pp. 108–121, 2014.
[53] I. Jain, V. K. Jain, and R. Jain, "Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification," *Applied Soft Computing*, vol. 62, pp. 203–215, 2018.
[54] K.-H. Chen, K.-J. Wang, K.-M. Wang, and M.-A. Angelia, "Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data," *Applied Soft Computing*, vol. 24, pp. 773–780, 2014.
[55] Z. Zhu, Y.-S. Ong, and M. Dash, "Markov blanket-embedded genetic algorithm for gene selection," *Pattern Recognition*, vol. 40, no. 11, pp. 3236–3248. Nov, 2007.
[56] S. Salesi, G. Cosma, and M. Mavrovouniotis, "TAGA: tabu Asexual Genetic Algorithm embedded in a filter/filter feature selection approach for high-dimensional data," *Information Sciences*, vol. 565, pp. 105–127, 2021.
[57] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
[58] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. Cruz, "Quadratic programming feature selection," *Journal of Machine Learning Research*, vol. 11, pp. 1491–1516, 2010.
[59] X. V. Nguyen, J. Chan, S. Romano, and J. Bailey, "Effective global approaches for mutual information based feature selection," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 512–521, New York, New York, USA, August 2014.
[60] O. Ludwig, U. Nunes, R. Araújo, L. Schnitman, and H. A. Lepikson, "Applications of information theory, genetic algorithms, and neural models to predict oil flow," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 2870–2885, 2009.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9325803, 11 pages
https://doi.org/10.1155/2022/9325803

Research Article

Small-Scale and Occluded Pedestrian Detection Using Multi Mapping Feature Extraction Function and Modified Soft-NMS

Addis Abebe Assefa,[1] **Wenhong Tian**,[1] **Kingsley Nketia Acheampong**,[1] **Muhammad Umar Aftab**,[2] **and Muhammad Ahmad**[2]

[1]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China*
[2]*Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan*

Correspondence should be addressed to Muhammad Ahmad; mahmad00@gmail.com

Received 3 January 2022; Revised 28 June 2022; Accepted 14 September 2022; Published 11 October 2022

Academic Editor: Anastasios D. Doulamis

In autonomous driving and Intelligent transportation systems, pedestrian detection is vital in reducing traffic accidents. However, detecting small-scale and occluded pedestrians is challenging due to the ineffective utilization of the low-feature content of small-scale objects. The main reasons behind this are the stochastic nature of weight initialization and the greedy nature of non-maximum suppression. To overcome the aforesaid issues, this work proposes a multifocus feature extractor module by fusing feature maps extracted from the Gaussian and Xavier mapping function to enhance the effective receptive field. We also employ a focused attention feature selection on a higher layer feature map of the single shot detector (SSD) region proposal module to blend with its low-layer feature to tackle the vanishing of the feature detail due to convolution and pooling operation. In addition, this work proposes a decaying nonmaximum suppression function considering score and Intersection Over Union (IOU) parameters to tackle high miss rates caused by greedy nonmaximum suppression used by SSD. Extensive experiments have been conducted on the Caltech pedestrian dataset with the original annotations and the improved annotations. Experimental results demonstrate the effectiveness of the proposed method, particularly for small and occluded pedestrians.

1. Introduction

Pedestrian detection is a fundamental task in computer vision applications such as surveillance, robotics, and automotive safety. Specifically, pedestrian detection in transportation is significant because it can save countless lives [1]. Despite extensive research on pedestrian detection, new studies show significant advances, signaling that a maximum threshold has yet to be reached, i.e., small-scale pedestrian detection and occlusion are two of the current state-of-the-art constraints. Insufficient feature strength in small objects and the stochastic nature of kernel initialization are the leading cause of incapability in detecting small objects [2, 3].

Occlusion is another challenging issue in pedestrian detection because it is difficult to compromise miss rate and accuracy when detectors are sensitive to the nonmaximum suppression (NMS) threshold in crowded environments [4]. The technology for detecting pedestrians is advancing all the time. Although the occlusion problem can be solved, there is still a significant barrier to overcome.

Videos taken by businesses such as banks and shopping malls obscure the majority of people on the street [5]. When pedestrians are hidden by background clutter or other objects, it can be more difficult to detect them. In the field of smart cities, pedestrian detection under occlusion has become a popular method of tracking people. Occupied pedestrian detection is useful in a wide range of fields, such as automated driving, intelligent video surveillance, robotics, human-computer interaction, and security. Assisted driving and self-driving vehicles are two of the most important

aspects of intelligent transportation. It is essential to detect humans even when they are partially obscured by objects. Drivers need to be aware of pedestrians and give them the benefit of the doubt when it comes to pedestrian detection under occlusion.

There are four levels of occlusion between pedestrians [6]: zero, one to 35%, 35 to 80%, and above 80%. Depending on the detection framework, occupied pedestrian detection can be broken down into two approaches: (1) traditional methods [7, 8] and (2) deep learning methods [9–12].

Traditional methods for dealing with occluded pedestrians include combining hand-engineered kernel features such as the Histogram of Oriented Gradient (HOG) descriptor [8], Scale Invariant Feature Transform (SIFT) [13], and aggregated channel feature [14] with linear support vector machine [15] or random forests [16]. The ability of HOG and SIFT to represent distinguishing characteristics of a pedestrian makes them popular algorithms. Many pedestrian detection frameworks have been presented to address a wide range of detection challenges, from the occluded to the visible and small to large scale.

Dollar et al. [17] presented Integral Channel Feature (ICF), which uses integral images to extract features from HOG and LUV color channels (HOG + LUV) and employs boosted decision forests for pedestrian detection to boost detection accuracy. Moreover, Dollar et al. developed three opposing cascaded modules (soft cascade, excitatory cascade, and inhibitory cascade) to maximize the inference rate. After ICF, handcrafted features with improved LUV color channels (Aggregated Channel Features (ACF), Rotated Filters [18], Locally Decorrelated Channel Features (LDCF) [19], Checkerboards [20], and SquaresChntrs [21]) were used to improve low-resolution image detection and enable detector invariance to changes in lighting conditions.

The alternative approach to tackle occluded pedestrians is the component-based method. Even if a portion of the pedestrian to be detected is obscured, the remaining parts can be used to determine the pedestrian's location. According to Leibe et al. [22], pedestrian detection algorithms in crowded scenes are equivalent to the prototype of pedestrian detection under occlusion. A key component of their approach is the use of probabilistic top-down segmentation to combine local and global cues. To better deal with pedestrians, Mohan et al. [23] found that pedestrians can be divided into four sections: head and shoulder, leg, left hand, and right hand. It is more effective to deal with occlusion. Occlusion-aware pedestrian detection frameworks were devised using a deformable part-based model (DPM) [24], which generalizes global appearance from local appearance. The first method produces acceptable detection results with minimal computational effort. Despite this, it has a high miss rate, has difficulty determining the proper aspect ratio between image pyramids and sliding window step size, and kernels are not learnable, meaning they must be constructed manually.

The second approach is based on deep learning and significantly boosts detection accuracy and response time. The approach to deep learning is primarily divided into two categories. First, there is the two-stage detector algorithm, which separates target recognition and location into two parts. For example, the family of recurrent convolutional neural networks (R-CNNs) has been used in [25, 26] for object detection. R-CNN is one of the most prominent deep learning-based object detection frameworks. Since then, the R-CNN family has been subjected to various studies to increase detection accuracy and response time. A super-resolution algorithm, a combination of handcrafted features with convolutional neural network (CNN) feature maps [11, 27–31] and CNN's layer fusion are the leading approaches built on top of the R-CNN family to deal with low-resolution, occlusion, and normal (fully visible and large scale) pedestrian detection.

Tian et al. [32] proposed the Deep-Parts, using Deep Part, they can automatically select critical body parts for occlusion handling from a part pool that includes parts of all sizes. An ensemble of detectors is learned and the output of the ensemble is integrated as a strategy for dealing with occlusion in these methods. However, the procedure is difficult and extensive. Furthermore, by combining a faster R-CNN with an attention mechanism composition [33], this method has a minimal training burden, making it easy to train. The overhead for this approach is minimal, making it simple to train. The use of attention mechanisms in CNN object detection has been widespread. In order for the detector to focus more on visible body parts, an additional mechanism has been added. The primary method for dealing with small-scale or low-resolution pedestrian detection is to increase the feature's resolution.

Tesema et al. [34] blends handcrafted features with convolutional feature maps to detect a low-resolution pedestrian. Chu et al. [35] synthesized high-level features using features from all levels. By adaptively merging multilayer features, Liu et al. [36] suggested a gated feature extraction module. Wang et al. [37] proposed a multiscale area proposal network with a decision forest for categorization to deal with scale-invariance differences. Inspired by the human detection and recognition mechanism, attention models have been developed and integrated on R–CNN for different tasks. For example, Zhang et al. [38] observed that individual streams depict different regions of an object and employed channel-wise attention [39] to detect occluded pedestrians. The maximum detection accuracy is achieved by all pedestrian detection frameworks built on top of the R-CNN family. However, they are complicated, with many moving parts, and can only be used for offline deep learning-based object detection.

Second (class-aware), it is a one-stage detector algorithm that includes the single shot detector (SSD) [40] and You Only Look Once (YOLO) [12, 41] object detection framework has been developed to address these concerns. The SSD object detector is entirely end-to-end, has no complex moving parts, and can perform in real time. The current state-of-the-arts built on top of class-aware frameworks yields a good result in the best conditions, such as a pedestrian with a comparable scale (above 80 pixels height) and complete visibility. However, detection and recognition capabilities for small-scale pedestrian and partial occlusion are severely compelled.

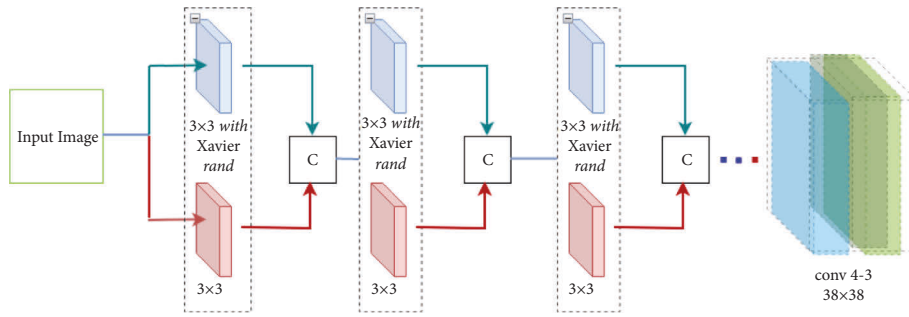


FIGURE 1: Our proposed backbone module. It has two branches i.e., the bottom branch is a sample layer of the VGG-16, and the top component is a new integrated layer in which the kernel size is equal in size to the bottom one except for weight initialization which is Xavier. At the end of each layer, feature maps are extracted from both branches, and combined **C** in-depth wise.

The poor performance of the state-of-the-art works in detecting small-scale or obstructed pedestrians is primarily due to the stochastic nature of kernel initialization, fading of an object resolution as feature maps traverse through deep layers, and the greedy nature of the SSD's nonmaximum suppression approach. The SSD begins with a base network (VGG-16). Later, convolution and pooling layers are replaced with a set of new convolution layers. The output fully connected (FC) layer is connected to each CONV layer. The most common complaint of SSDs is that they do not function well for small items because small objects do not always exist on all feature maps. Hence, the curse of this approach for small-scale pedestrian detection is that there is a high probability of losing all of the features of an object, i.e., reduction of resolution.

In other words, SSDs do not function well for small items because small objects' attributes do not always exist on all layers of the region proposal module. Besides, occlusion creates a challenge in detecting pedestrians, especially in a crowded situation. In the Caltech pedestrian dataset [5], for example, notes that other pedestrians occlude 70 percent of pedestrians. It is challenging to compromise miss rate and accuracy when detectors are sensitive to the threshold of nonmaximum suppression (NMS) in crowded settings.

We propose a solution to overcome the challenges mentioned above; we offer a new architecture with a channel-aware attention feature fusion and nonmaximum suppression that considers score and IOU parameters to reduce a miss rate and increase accuracy. Our architecture adopts VGG-16 [42] by adding a new branch with the same kernel size as the original architecture. However, we employ Xavier [43] weight initialization, which diffuses the concentration in the center out to the periphery, unlike normal Gaussian distribution [44] used by VGG-16 weight initialization. So, a combination of features from both branches can magnify the effective receptive field. We also propose a channel-aware attention feature fusion between each CONV layer of the region proposal module, as shown in Figure 1, to overcome the vanishing of features on a small scale. In addition, to handle pedestrians with occlusion, we propose nonmaximum suppression which is a modified version of Soft-NMS. Specifically, we consider the difference between the objects proposals score parameter and an intersection (IOU) used by soft-NMS to reduce the miss rate caused by the greedy nature of nonmaximum suppression. The main contributions made in this work are as follows:

(1) We propose a framework that incorporates channel-aware feature fusion with a heterogeneous mapping function to efficiently employ all features in a receptive field while avoiding vanishing gradients generated by a small-scale object.

(2) We explore how the field of view of a unit value in a particular layer in the network is affected by the kernels' weight initialization. Our experiments show that Gaussian random weight initialization assigns a large scale at the center of the kernel and a small value for the outermost; as a result, only the central receptive field has a high impact on unit out. We find that equal levels of the relevance of features within the receptive field enable a framework to attain maximum effective receptive field without a shallow layer.

(3) We find that applying greedy nonmaximum suppression to crowded pedestrian detection reduces precision by excluding detection with high scores that are likely to be false positives. To overcome these limitations, we employ an adaptive bounding box suppression.

(4) Experiments and analyses are conducted on the Caltech pedestrian dataset [5] with the original and new annotations. Our approach achieves the miss-rate (MR) of 9.1 and 6.08, outperforming detection with occlusion performance.

The remainder of this article is organized as follows: Section 2 goes through the related work. Section 3 goes through the proposed architecture and its components in detail. In Section 4, we conduct experiments, analyze the result, and conduct an ablation study. Finally, we sum up our work and make some recommendations for further research.

2. Related Works

So far, pedestrian detection studies have yielded positive results in ideal conditions such as large scale, full visibility, and high resolution. Despite its great success, pedestrian detection still faces a number of challenges under certain conditions, as discussed below.

2.1. Scale Aware Pedestrian Detection. The current state-of-the-art in small-scale pedestrian detection methods focuses on enhancing weak signals in small-scale objects and preserving that signal while applying feature extraction operations such as convolution and pooling. Hybrid channel fusion and pure CNN feature maps are the two main approaches to small-scale pedestrian detection. They merged the advantages of handmade and CNN kernels in hybrid approaches, integrating high-resolution, and easy-to-compute handcrafted feature channels with low resolution and computationally expensive CNN feature channels.

On the other hand, some approaches use the handcrafted feature for region proposals. CNN for categorization applies CNN for bounding boxes generation while based on the extracted feature shallow classifier. The handcrafted filtered channel features (FCF) [20] were proposed to be replaced by convolutional channel features (CCF), in which each pixel in the final convolutional layer is treated as a single feature by Yan et al. [45].

Hu et al. [46] trained an ensemble of boosted decision forests using features from different layers of a CNN. Zhang et al. [47] and Tesemaa et al. [34] used the region proposal network (RPN) as an initial pedestrian detector, then trained a shallow classifier with in-depth features to refine detection results. Li et al. [48] proposed extracting multi-resolution in-depth features from different convolutional networks. Sheng et al. [48] created a filtered channel framework that combined deep semantic segmentation features with shallow handcrafted channel features. Wang et al. [37] created a multi-scale region proposal network that included a decision forest for classification to deal with scale differences.

The other approach is detecting objects on multiple scales [21] suggests a cohesive multiscale CNN (MS-CNN) that detects objects at multiple intermediate layers and up samples to avert feature map resolution issues when dealing with small instances. Instead of a single downstream classifier, the fused deep neural network (F-DNN + SS) method [49] employs a derivation of the faster R-CNN framework fusing multiple parallel classifiers using soft-rejection, such as Resnet and Google net, as well as pixel-wise semantic segmentation in postprocessing to suppress background proposals. The other problem is related to pose and learning method, for example, Wiedemer et al. and Tang et al. [50, 51] demonstrate that the combination of supervised domain adaptation with fine-tuning is effective in tackling settings posses caused by geometric distortions, perspective changes, and the scarcity of training samples. This paper suggests a multifocus feature extractor base module that combines feature maps extracted from the Gaussian and Xavier mapping functions in order to improve the effective receptive field which results in a signal of small-scale object to be used effectively.

2.2. Occlusion Aware Pedestrian Detection. The occlusion of pedestrians is a typical problem. For instance, 53% of pedestrians are obscured in the Caltech pedestrian dataset. The two types of pedestrian occlusions are interclass and intraclass occlusions. When other objects obstruct pedestrians, this is known as interclass occlusion (not pedestrians). Intraclass occlusion occurs when other pedestrians obstruct pedestrians. To handle interclass occlusion, emphasize the unoccluded part's characteristics while repressing the occluded part's attributes. To cover occlusion patterns [32, 52, 53], for example, learn different parts of body detectors. The challenging part of this multipart detector is combining the results. For the proposed model, taking into account the aspect ratio of a human's height and width, we structure a region proposal scheme.

The algorithm generates multiple bounding boxes with varying scales and aspect ratios, each centered on a single pixel. This configuration of human-based anchors enables the model to perform exceptionally well when determining whether a particular type of pedestrian appears in an image. We also include a small aspect ratio with equal height and width to enable the model to detect a pedestrian that is visible only above the neck. In addition, in the Caltech pedestrian dataset [5], the occlusion statistics indicate the number of occlusions caused by a pedestrian (intraclass). Therefore, the preceding setup enables the proposed model to predict a higher detection precision for intraclass occlusion.

Annotating the visible part of a body [54–59] is the other method; this technique consumes resources for annotation. NMS must remove duplicate bounding boxes when detecting pedestrians. As an adverse effect, NMS merges the bounding boxes of different pedestrians in a crowd scene (intra-class occlusion). Improved NMS strategies, such as dynamic NMS threshold, are one solution to this problem. To combat high miss rates caused by greedy nonmaximum suppression, we propose an adaptive bounding box suppression that takes score and Intersection Over Union (IOU) parameters into account.

3. The Proposed Architecture

This section introduces our proposed architecture, its components, a few fundamental design principles, and experiments and analysis. The proposed framework incorporates three submodules: the backbone module, the dense multiscale region proposal module, and the adaptive bounding box suppression module. The complete pipeline is presented in Algorithm 1.

3.1. The Backbone Module. The backbone-network extracts salient features from the input image. Using the pretrained ImageNet [60] model as the basis network offers multiple benefits. The benefit includes the flexibility to incorporate several cutting-edge models, the elimination of the need for a powerful machine to train, and a faster training period. However, the ImageNet dataset contains objects of "medium" size, spanning from 60 to 140 pixels, outside the intended domain, such as when a dataset is in small scale or occlusion. In addition, A single-scale kernel feature extractor like VGG16 has few parameters compared with a multiscale feature, but the signal strength at the end of a layer is low when compared with a multiscale feature.

As a result, we propose a new architecture that adopts VGG-16 by appending a new branch layer with the same kernel size as the original architecture. However, we apply Xavier [43] weight initialization, which diffuses the concentration in the center out to the periphery, unlike normal Gaussian distribution [44] used by VGG-16 weight initialization. Our base network is a multi-feature extraction model like Google net [61], mobileNet [62], and ResNet [63]. They can capture salient features of an image, enabling us to get a maximum effective receptive field with a shallow layer and enabling the region proposal regression module to get a sufficient signal. However, they contain many parameters when compared with our proposed base network. Finally, we combine feature maps from an integrated branch layer and original feature maps into channel dimensions as shown in Figure 1.

3.2. Dense Multiscale Region Proposal Module. This module shares a similar structure with the SSD region proposal module. The SSD detection framework uses the VGG-16 Conv4-3 feature map as its first scale, which has a scale of a different feature compared to other layers. The remaining five scales of the SSDs framework are generated by applying convolution operations on VGG-16 Conv5-3 feature maps. Our experiment found that the feature detail of a small-scale object does not propagate through all hierarchical scales with such a framework.

The vanishing of the feature detail results in poor performance causing hurting bounding box learning, and the bounding boxes proposal layer has a high miss rate. Hence, to tackle this limitation we apply a focused attention selection mechanism [64, 65] on the VGG-16 Conv4-3 layer to blend with the Conv5-3 feature, that preserves a long-term dependence between layers. It results in the propagation of more detailed information about an object to a layer that contains semantic information while avoiding background clutter and semantic ambiguity. All hierarchies of the feature layer receive feature maps of their previous layer which makes an object's feature pass-through all layers. Figure 2 shows how a lower-layer feature maps integrated with a higher layer feature. In this combination, a feature map of a Conv5-3 pass-through attention score (FE) is learned by 1×1 kernel as follows:

$$\mathrm{FE} = W_g * Y, \tag{1}$$

where $*$ denotes a convolution operation, W_g is the learned weight matrices implemented as a 1×1 convolution, and Y is the feature map of conv5-3. Then, the soft-max normalization is applied on a value of *FE* to get the feature selector, which assures non-negative selection of feature map.

$$\mathrm{SM}_{i,j} = \frac{\exp^{\mathrm{FE}_{i,j}}}{\sum_i^W \sum_j^H \exp^{\mathrm{FE}_{i,j}}}, \tag{2}$$

where, $\mathrm{SM} \in R^{W \times H \times 1}$ and $S_{i,j}$ are the score at position (i, j). A new lower-layer features is given by $X^s_{i,j,c} = SM_{i,j} * X_{i,j,c}$. This operation gives a more relevant lower-layer feature. Here, $X_{i,j,c}$ denote the value in X with a spatial location (i, j) at channel c.

There are six hierarchical convolution layers in the region proposal module. To improve layer feature map resolution and enable minor object features to reach the end of the feature proposal region, we combine feature maps from higher layers with feature maps from lower layers so that they can be detected. Background features propagate forward due to direct integration of layer's feature maps, reducing detection accuracy and increasing miss rate. As a result, we only use the feature selector to forward important feature maps from the lower layer to the higher layer.

This feature selector assigns attention scores to each local position on the low-layer feature map, indicating the low-layer features' importance. The attention score selector is learned by $1 \times 1 \times c$ kernel size, where c is the feature map depth. As shown in Figure 1, the first and last layer of the region proposal module is directly connected. However, the middle layers feature map is fussed, so based on this configuration the number of parameters is as follows: $1 \times 1 \times 512$ kernel for the second layer and the remaining three layers each has $1 \times 1 \times 256$ kernel size. Hence the total number of parameters is 1280.

The SSD's bounding boxes' width and height are defined by the aspect ratios (1, 0.5, 2, 0.333, and 3) with scales starting at 0.2 and growing linearly to the rightmost layer at a scale of 0.9. However, this design distribution does not best fit a common pedestrian dataset [5] because the equation [40] of the SSD's bounding boxes definition yields 50% of boxes with a width size greater than height resulting in misalignment between anchors and ground-truth bounding box features. As a result, we define bounding boxes shape as illustrated in the following equation:

$$\begin{aligned} \mathrm{Height} &= \mathrm{scale} \times \mathrm{aspect\_ratio} \times \mathrm{feature\_map\_height}, \\ \mathrm{Width} &= \mathrm{scale} \times \mathrm{aspect\_ratio} \times \mathrm{feature\_map\_width}. \end{aligned} \tag{3}$$

There is no standard for defining the number of model anchors and their characteristics. Nevertheless, a good decision is based on several factors. The problem type (the density of the objects in the dataset) i.e., are the objects in the dataset sparse or dense? Or if the objects in the image are large, small, or mixed and the smallest and largest boxes to be detected in the dataset are common considerations. Moreover, CNN's key characteristics of multiscale, regular deformation, and sparseness make it the optimal view, particularly for our problem of dealing with the dynamically changing nature of crowd density and the scale of various objects.

First, the bottom layers of CNN aggregate very fine, small-scale information, which is close neighbor information, and then, through the cascading of convolutional and subsampling operations, gradually aggregate information on a larger scale. Consequently, our anchor number and aspect ratio configuration are assigned based on these and the preceding factors. There are so many anchors for the lower layer because this layer's feature contains many objects, and as a layer's height increases, we reduce the number of anchors, because the feature may contain a few objects with

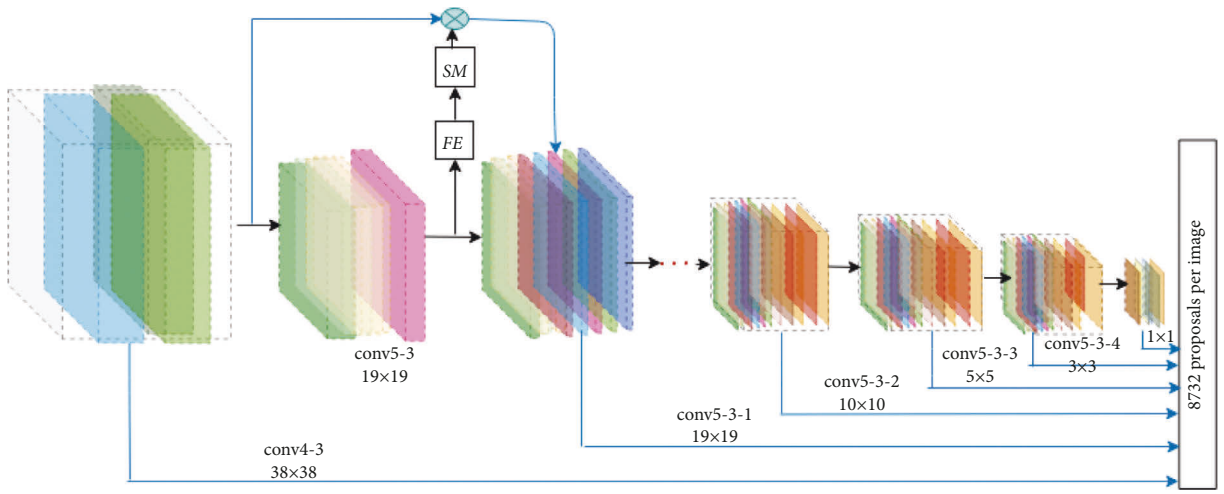


Figure 2: Our proposed region proposal module, here VGG-16 Conv5-3 layer feature map pass-through attention score feature extractor module FE, which is learned by 1×1 kernel. The output from FE pass-through the soft-max normalization SM to ensure a non-negative feature selector. Finally, the output of SM is multiplied with a low-layer feature map which enables a significant feature from the low-layer feature maps to pass through the higher layer, i.e., can avoid background jumble and semantic ambiguity.

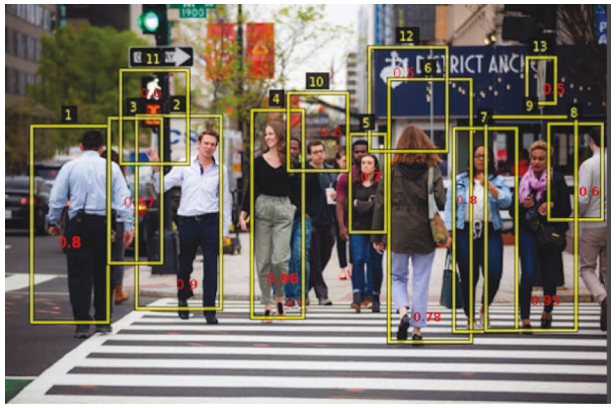


Figure 3: Our figure shows a pedestrian's detection score; with the help of the declining score function, occluded pedestrians can be detected with a low score value. When greedy-NMS is used, a pedestrian with label 9 overlaps with labels 7 and 8, results in a 0 score for both. The penalization of the score, on the other hand, aids in the detection of the occluded pedestrian.

large dimensions. Consequently, given the preceding factors, the configuration of the aspect ratio (1, 0.5, 3, 4, and 2) and the number of anchors (6, 6, 4, 4, and 4) at each layer of a region proposal yielded a satisfactory outcome after several experiments.

3.3. Adaptive Bounding Box Suppression Module. To demonstrate the effect of nonmaximum suppression (NMS) on occluded pedestrian detection, we examine various NMS techniques, such as greedy-NMS [66], soft-NMS [67], and our proposed modified soft-NMS called score-soft NMS. A greedy-NMS applies a hard threshold when deciding what should be retained or eliminated from K's bounding box neighborhood. Setting maximum and minimum threshold values has a cost, for example, suppressing all nearby detection boxes with a low threshold value increases the miss rate.

Furthermore, using a high threshold value would result in more false positives, lowering average precision. Figure 3 depicts the effect of greedy-NMS on a crowded pedestrian. A pedestrian with a label of 2 has neighbors labeled with 3 and 11; their score is 0.67 and 0.6, respectively. Because both boxes overlap significantly with label 2, greedy-NMS suppresses them and assigns a score of zero. While Soft-NMS decays the scores of 2's neighborhood, the score for detection boxes with a higher overlap with 2 should be decayed more, as they are more likely to be false positives. The time complexity of this algorithm is equivalent to that of soft-NMS, but the decaying function is different where $f(\mathrm{iou}(M, b_i), \mathrm{diff}(s_m, s_i))$ is the weighting function based on overlap. In each iteration of removing an anchor has a computational complexity of $O(N)$, where N is the number of detection boxes.

Soft-NMS outperforms greedy NMS in terms of precision, especially when dealing with crowded pedestrians. However, the rescoring function based on the overlap parameter results in a low decay rate for a proposal having a small score value. Hence, to tackle this, we modified the soft-NMS of rescoring function by considering a proposal's score and overlap. For example, detection boxes with higher overlaps with 2's and a low objectness score decay more than the exact overlaps but with a high objectness score.

As shown in equation (4), decaying is an exponential function of the difference between a proposal having a high score and its neighbor's scores with their overlap value. This results in a detection box having a higher overlap and a lower objectness score with the highest score proposal decaying faster than a high score value. This update rule is applied every cycle, and the scores of all remaining detection boxes are updated. The algorithm's top, middle, and bottom boxes represent greedy-NMS, soft-NMS, and modified soft-NMS, respectively.

$$s_i = e^{-(\mathrm{diff}(s_m, s_i))} s_i e^{-(\mathrm{iou}(M, b_i))}, \quad (4)$$

where $\forall b_i \notin D, s_i$ is the revised abjectness score of a proposal. The overlap-NMS technique is depicted in full in the algorithm, which includes the overlap and abjectness score-based weighting function.

4. Experiments and Results

4.1. Dataset and Evaluation Metrics. To evaluate the usefulness of the proposed approach, we used the Caltech dataset [5], which is a benchmark for testing pedestrian detection algorithms. The dataset is significant and difficult to evaluate, comprising around 10 hours of videos (at 30 frames per second) taken from a vehicle traveling through metropolitan traffic. In every frame of the raw Caltech dataset, the bounding boxes of pedestrian instances have been heavily annotated. In 250,000 frames, 350,000 bounding boxes with about 2,300 distinct pedestrians are classified. We employ the log-average miss rate in log-space in the range [0.02 to 1] to describe detector performance. Pedestrian height and proportion of occlusion have been used to construct various assessment settings. The following are the parameters utilized in this paper.

4.1.1. Setups Involving Scale. Near (80 pixels or more), medium (30 pixels to 80 pixels), and far (30 pixels or less) are the three categories of pedestrian distribution in the dataset; the medium size is excellent for automobile systems. It is the most often used evaluation setting, and practically all pedestrian detection research studies use it as a standard evaluation benchmark. In this study, it is also the default setting for evaluation.

4.1.2. Setups Involving Occlusion. The partial and heavy occlusion categories apply to pedestrians between 30 pixels to 80 pixels and have 1–35 and 36–80 of their body parts occluded, respectively. We test our methods on the updated annotations supplied by [18], which amend the mistakes in the original annotations and the Caltech dataset with the original annotations.

4.1.3. Log-Average Miss Rate (LAMR). Used to measure the performance of the object detector by comparing the log-average miss rate of the detection results compared to ground-truth data.

$$\mathrm{LAMR} = \exp\left(\frac{1}{p}\sum_{f} \log\left(mr\left(\mathrm{argmax}_{fppi(c)\le f}\, fppi(c)\right)\right)\right), \quad (5)$$

where, $fppi$ is the number of false positives per image, p is the number of $fppi$ reference points, and c is a confidence threshold c which is used as a control variable. By decreasing c, more detections are taken into account for evaluation resulting in more possible true or false positives, and possibly fewer false negatives, and mr is the miss rate which is equal to the division of the number of false negatives $fn(c)$ by the sum of the number of true positives $(tp(c))$ and the number of false positives $(fp(c))$. For each $fppi$ reference point the corresponding mr value is used. In the absence of a miss-rate value for a given f the highest existent *fppi* value is used as a new reference point.

4.2. Ablation Study. First, we conduct ablation experiments to evaluate the single-scale kernel properties based on the features produced from the VGG-16 base framework. We use recall rates at various IoU thresholds for evaluation. The results of the experiment are shown in Table 1. The Conv5-3-4 feature has the highest MR of 13. The impact of low-resolution features on proposal quality can also be noticed. MR increases as we progress from low-level to high-level features. Conv4-3 begins to decline, which may be attributed to a diminished depiction of the shallower layers. We contend that the low-resolution features are the reason for the poor performance.

To demonstrate this, we employ our modified VGG-16, multiscale feature extraction-based network (on the same set of region proposals). Table 1 of column 6 shows the MR result after modification of the VGG-16 base network. From the result, we can see that the MR of each layer of feature maps is enhanced. This fact reveals that multiscale feature extraction is vital for an effective proposal because it is large enough to capture the prominent characteristics of an image, allowing for a maximum effective receptive field with a shallow layer and sufficient signal reach at the base network's end layer.

4.3. Evaluation of the Dense Multiscale Region Proposal Detector. This section is devoted to exploring the effectiveness of the proposed dense multiscale region proposal detector. The feature maps from Conv4-3 with a resolution 38×38 of VGG-16 are our framework's first layer because they achieve the best performance among the different layers [40]. Then, using a focused attention selection mechanism [68], we blend the higher layer with the lower-layer feature to generate the remaining five hierarchical feature maps.

Table 2 shows the results of our proposed framework for all combinations of feature maps. The results show that the combination of features outperforms each CNN feature alone, such as SSD feature representation. The combination of Conv-4-3 and Conv 5–3 layer of VGG-16 achieves an MR of 9.1, which is slightly better than the best MR of 9.6 achieved in [34]. In their work, the best result is achieved by the combination of conv3-3 and the trous version of Conv4-3 takes extra computation time to recompute the conv4-3 features maps with the trous trick and adds background clutter and semantic ambiguity to a higher layer. While attention selection feature integration requires more computation time, it allows lower-layer feature maps to propagate forward if they have a high correlation with higher-layer feature maps.

As a result, more detailed information about an object is flowed to a semantic layer, avoiding background clutter and semantic ambiguity. This suggests that optimizing the

Input: $B = b_1, \ldots, b_N$, $S = s_1, \ldots, s_N$, N_t
B is the list of initial detection boxes
S contains corresponding detection scores
N_t is the NMS threshold
$D \leftarrow$ **while** $B \neq$ empty **do**
  $m \leftarrow$ argmax S
$M \leftarrow b_m$
  $D \leftarrow D \cup M$
  $B \leftarrow B - M$
  for b_i in B **do**
    if iou $(M, b_i) \geq N_t$ **then**
      $B \leftarrow B - b_i$,
      $S \leftarrow S - s_i$
    $s_i \leftarrow f(\text{iou}(M, b_i))$
    $s_i \leftarrow f(\text{iou}(M, b_i), \text{diff}(s_m, s_i))$

Algorithm 1: The algorithm shows the modified version of Soft-NMS. The last box explicitly shows our proposed objectness score decaying function, a function of IOU and score parameter. The function revise the detection scores by scaling them as Gaussian function of overlap and objectness score.

Table 1: Comparison between the base network VGG-16 and our proposed modified base module for a small-scale pedestrian on each hierarchy of layers.

Feature map	Resolution	Base network	MR	Base-network	MR
Conv4-3	38×38	VGG-16	9.6	Modified VGG-16	9.1
Conv5-3	19×19	VGG-16	10.2	Modified VGG-16	10
Conv5-3-1	10×10	VGG-16	10.8	Modified VGG-16	10.3
Conv5-3-2	5×5	VGG-16	11.5	Modified VGG-16	11.5
Conv5-3-3	3×3	VGG-16	12.8	Modified VGG-16	12.3
Conv5-3-4	1×1	VGG-16	13	Modified VGG-16	12.6

effective receptive field of a base network and using an efficient feature integration algorithm can provide quality object proposals and improve detection accuracy.

4.4. Evaluation with respect to Occlusion. To demonstrate the importance of score-aware soft-max suppression in occlusion-aware pedestrian detection, we substitute Base − Mo de l + Region − Proposal of SSD greedy-NMS with soft-NMS and our suggested NMS. We investigate the applicability of our technique in various occlusion evaluation circumstances (none, partial, and heavy). Table 3 shows the outcomes. This comparison is carried out using the MR metric, applied to all feature map hierarchies of the region proposal module. According to this analysis, the considerable improvement of nonmaximum bounding box suppression is crucial to enhancing the accuracy of occluded detection. It demonstrates one approach to dealing with occlusion aware-detection systems.

4.5. Comparison with State-of-the-Art Methods on Caltech with the Original and New Annotations. Table 4 compares our best results of dense multiscale feature extraction with score-soft pedestrian detection (DMSSPD) to state-of-the-art methods on the Caltech dataset with original and latest annotations. On the two benchmarks, our technique produced MR of 9.1% and 6.08%, respectively. Several research classes have produced better results than ours, but they all employ more complex structures and need higher processing costs.

For instance, the work [69] achieved an outstanding result. A single-shot convolutional multi-box detector for pedestrian candidates a classification system employing the concept of ensemble learning to improve the detection accuracy and classify the generated candidates, a novel soft-rejection fusion method to assign floating point labels to the generated pedestrian candidates, and a deep context aggregation semantic segmentation network that provides the pixel-level classification of the scene.

The aforementioned components make the model more computationally complex than our proposed method because our proposed method has three moving parts: a single-shot convolutional multi-box detector for pedestrian candidates, which is identical to [69]; a focused attention selection module with a complexity of 11c kernel size where c is the depth of six layers for propagating essential features to the next layer; and an adaptive bounding box suppression module is the weighting function. The computational complexity of each step is $O(N)$, where N is the number of detection boxes. Furthermore, [72] utilized a multitask infusion framework for pedestrian detection and semantic segmentation joint subprediction.

The segmentation infusion layer yields more refined shared feature maps, which tend to illuminate pedestrians and facilitate pedestrian detection in a subsequent step. It consists of four elements, including weak segmentation supervision, proposal padding, cost-sensitive weighting, and

Table 2: The detection performance (MR) of the region proposal module layers for medium and small-scale pedestrians on average.

SSD feature map	Base network	Resolution	MR	Modified SSD feature map	MR
Conv4-3	VGG-16	38×38	11.5	Conv4-3	11.10
Conv5-3	VGG-16	19×19	12.32	$A \leftarrow conv4-3, conv5-3$	10.8
Conv5-3-1	VGG-16	10×10	12.65	$B \leftarrow conv5-3-1, A$	10.9
Conv5-3-2	VGG-16	5×5	13.02	$C \leftarrow conv5-3-2, B$	12.1
Conv5-3-3	VGG-16	3×3	14.36	$D \leftarrow conv5-3-3, C$	13.01
Conv5-3-4	VGG-16	1×1	16.2	Conv5-3-4, D	13.98

Table 3: The effect of nonmaximum suppression under different occlusion settings and detection performance (MR) under different scales.

NMS	Detection performance (MR) under different scales and occlusion									
	VGG-16 + RPN					Proposed method				
	Nea	Med.	Part	Hea	NoN	Nea	Med	Part	Hea	NoN
G-NMS	58.09	2.59	78.77	27.64	11.33	58.09	2.23	77.77	78.8	27.64
Soft-NMS	56.31	2.59	76.62	26.54	9.95	56.31	2.32	72.62	24.54	9.11
Score-soft-NMS	54.35	1.66	74.72	20.4	8.15	54.35	1.12	71.03	22.8	8.23

Table 4: Comparison of our works with other works on Caltech with the original (MR^O).

Methods	MR^O	MR^N
F-DNN [69]	8.65	—
HCD [34]	9.53	6.41
PDM [70]	10.2	—
Hyper learner [71]	—	5.5
SDS-RCNN [72]	7.36	6.44
CompACT-deep [73]	11.75	9.15
MCF [74]	10.40	7.98
TLL-TFA [75]	7.39	—
DMSSPD	9.1	6.08

stricter oversight. Because each pixel in the image is computed to identify a foreground feature, each component has numerous parameters and computational complexity.

5. Conclusion

The detection of pedestrians is a primary concern. In ideal settings, such as a pedestrian with a comparable scale (above 80 pixels in height) and complete visibility, the current state-of-the-art produces an excellent outcome. Despite significant advancements, recognizing small-scale pedestrians remains a source of the problem. As a result, to address this issue, we present an average model weight ensemble framework that can learn a variety of mapping functions and allows features to have an equal impact within the receptive field by spreading the concentration of a sensory field center to the periphery. We also improve an adaptive bounding box suppression that maps objects to nearby points via dimensionality reduction.

Data Availability

The dataset is publicly available. We cite the dataset in Section 4 for anyone interested in using it.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. A. Assefa and T. Wenhong, "Bio-mechanical distracted driver recognition based on stacked autoencoder and convolutional neural network," in *Proceedings of the 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP)*, pp. 449–453, IEEE, Weihai, China, September 2019.

[2] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems*, pp. 253–256, IEEE, Paris, France, May 2010.

[3] Y.-T. Zhou, R. Chellappa, A. Vaid, and B. K. Jenkins, "Image restoration using a neural network," *IEEE Transactions on Acoustics, Speech, & Signal Processing*, vol. 36, no. 7, pp. 1141–1151, 1988.

[4] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, pp. 850–855, IEEE, Hong Kong, China, August 2006.

[5] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[6] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Proceedings of the European Conference on Computer Vision*, pp. 645–659, Springer, Heidelberg, Germany, June 2012.

[7] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Proceedings of the international conference on image processing*, IEEE, New York, NY, USA, September 2002.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pp. 886–893, IEEE, San Diego, CA, USA, June 2005.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, pp. 818–833, Springer, New York, NY, USA, June 2014.

[10] C. Ning, L. Menglu, Y. Hao, S. Xueping, and L. Yunhong, "Survey of pedestrian detection with occlusion," *Complex & Intelligent Systems*, vol. 7, no. 1, pp. 577–587, 2021.

[11] M. Ahmad, A. M. Khan, M. Mazzara, S. Distefano, M. Ali, and M. S. Sarfraz, "A fast and compact 3-d cnn for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[13] T. Lindeberg, "Scale Invariant Feature Transform," *Scholarpedia*, vol. 7, p. 10491, 2012.

[14] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[15] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer Science & Business Media, Heidelberg, Germany, 2008.

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

[17] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral Channel Features," in *Proceedings of the British Machine Vision Conference, BMVC 2009*, London, UK, January 2009.

[18] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *Proceedings of the iEEE Conference on Computer Vision and Pattern Recognition*, pp. 1259–1267, Las Vegas, NV, USA, June 2016.

[19] W. Nam, P. Dollár, and J. H. Han, "Local decorrelation for improved pedestrian detection," *Advances in Neural Information Processing Systems*, vol. 27, pp. 424–432, 2014.

[20] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," *CVPR*, vol. 1, p. 4, 2015.

[21] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3666–3673, Portland, OR, USA, June 2013.

[22] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 878–885, IEEE, San Diego, CA, USA, June 2005.

[23] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349–361, 2001.

[24] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[27] M. Ahmad, S. Shabbir, S. K. Roy et al., "Hyperspectral image classification - traditional to deep models: a survey for future prospects," *Ieee Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, no. 1–1, pp. 968–999, 2022.

[28] M. Ahmad, S. Shabbir, R. A. Raza, M. Mazzara, S. Distefano, and A. M. Khan, "Artifacts of different dimension reduction methods on hybrid cnn feature hierarchy for hyperspectral image classification," *Optik*, vol. 246, Article ID 167757, 2021.

[29] Y. Huang, J. Llach, and C. Zhang, "A method of small object detection and tracking based on particle filters," in *Proceedings of the 2008 19th International Conference on Pattern Recognition*, pp. 1–4, IEEE, Tampa, FL, USA, December 2008.

[30] D. I. Kosmopoulos, N. D. Doulamis, and A. S. Voulodimos, "Bayesian filter based behavior recognition in workflows allowing for user feedback," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 422–434, 2012.

[31] T. Zhang, C. Xu, and M.-H. Yang, "Learning multi-task correlation particle filters for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 365–378, 2019.

[32] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1904–1912, Santiago, Chile, December 2015.

[33] A. Sumi and T. Santha, *Frame Level Difference (Fld) Features to Detect Partially Occluded Pedestrian for Adas*, NISCAIR-CSIR, India, 2019.

[34] F. B. Tesema, H. Wu, M. Chen, J. Lin, W. Zhu, and K. Huang, "Hybrid channel based pedestrian detection," *Neurocomputing*, vol. 389, pp. 1–8, 2020.

[35] J. Chu, Z. Guo, and L. Leng, "Object detection based on multilayer convolution feature fusion and online hard example mining," *IEEE Access*, vol. 6, Article ID 19959, 2018.

[36] X. Liu, K.-A. Toh, and J. P. Allebach, "Pedestrian detection using pixel difference matrix projection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1441–1454, 2020.

[37] H. Wang, Y. Li, and S. Wang, "Fast pedestrian detection with attention-enhanced multi-scale rpn and soft-cascaded decision trees," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5086–5093, 2020.

[38] S. Zhang, X. Zhao, L. Fang, H. Fei, and H. Song, "Led: localization-quality estimation embedded detector," in *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 584–588, IEEE, Athens, Greece, October 2018.

[39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[40] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, New York, NY, USA, September 2016.

[41] H. Xu, M. Guo, N. Nedjah, J. Zhang, and P. Li, "Vehicle and pedestrian detection algorithm based on lightweight yolov3-promote and semi-precision acceleration," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2022.

[42] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, https://arxiv.org/abs/1409.1556.

[43] S. K. Kumar, "On Weight Initialization in Deep Neural Networks," 2017, https://arxiv.org/abs/1704.08863.

[44] D. G. Altman and J. M. Bland, "Statistics notes: the normal distribution," *BMJ*, vol. 310, no. 6975, p. 298, 1995.

[45] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li, "Object detection by labeling superpixels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5107–5116, Boston, MA, USA, June 2015.

[46] Q. Hu, P. Wang, C. Shen, A. van den Hengel, and F. Porikli, "Pushing the limits of deep cnns for pedestrian detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1358–1368, 2018.

[47] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in *Proceedings of the European Conference on Computer Vision*, pp. 443–457, Springer, Berlin, Germany, July 2016.

[48] C. Li, X. Wang, and W. Liu, "Neural features for pedestrian detection," *Neurocomputing*, vol. 238, pp. 420–432, 2017.

[49] Y. Cao, D. Guan, Y. Wu, J. Yang, Y. Cao, and M. Y. Yang, "Box-level segmentation supervised deep neural networks for accurate and real-time multispectral pedestrian detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 150, pp. 70–79, 2019.

[50] T. Wiedemer, S. Wolf, A. Schumann, K. Ma, and J. Beyerer, "Few-shot supervised prototype alignment for pedestrian detection on fisheye images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4142–4153, New Orleans, LA, USA, June 2022.

[51] Z.-R. Tang, R. Hu, Y. Chen, Z.-H. Sun, and M. Li, "Multi-expert learning for fusion of pedestrian detection bounding box," *Knowledge-Based Systems*, vol. 241, Article ID 108254, 2022.

[52] J. Xie, Y. Pang, H. Cholakkal, R. Anwer, F. Khan, and L. Shao, "Psc-net: learning part spatial co-occurrence for occluded pedestrian detection," *Science China Information Sciences*, vol. 64, no. 2, Article ID 120103, 2021.

[53] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Pedhunter: occlusion robust pedestrian detector in crowded scenes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, Article ID 10639, 2020.

[54] Y. Pang, Y. Li, J. Shen, and L. Shao, "Towards bridging semantic gap to improve semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4230–4239, Seoul, Republic of Korea, October 2019.

[55] Z. Chen, W. Ouyang, T. Liu, and D. Tao, "A shape transformation-based dataset augmentation framework for pedestrian detection," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1121–1138, 2021.

[56] N. Bakalos, A. Voulodimos, N. Doulamis et al., "Protecting water infrastructure from cyber and physical threats: using multimodal data fusion and adaptive deep learning to monitor critical systems," *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 36–48, 2019.

[57] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, "Object-centric auto-encoders and dummy anomalies for abnormal event detection in video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7842–7851, Long Beach, CA, USA, June 2019.

[58] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alexe, "Detecting abnormal events in video using narrowed normality clusters," in *Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1951–1960, IEEE, Waikoloa, HI, USA, January 2019.

[59] I. Katsamenis, N. Bakalos, E. E. Karolou, A. Doulamis, and N. Doulamis, "Fall detection using multi-property spatio-temporal autoencoders in maritime environments," *Technologies*, vol. 10, no. 2, 2022.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, Miami, FL, USA, June 2009.

[61] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, June 2015.

[62] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: Efficient Convolutional Neural Networks for mobile Vision Applications," 2017, https://arxiv.org/abs/1704.04861.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, June 2016.

[64] K. Morabia, J. Arora, and T. Vijaykumar, "Attention-based Joint Detection of Object and Semantic Part," 2020, https://arxiv.org/abs/2007.02419.

[65] S. Cao and L. Wang, "Attention Head Masking for Inference Time Content Selection in Abstractive Summarization," 2021, https://arxiv.org/abs/2104.02205.

[66] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. 20, no. 5, pp. 562–569, 1971.

[67] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5561–5569, April 2017.

[68] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," 2014, https://arxiv.org/abs/1409.0473.

[69] X. Du, M. El-Khamy, V. I. Morariu, J. Lee, and L. Davis, "Fused Deep Neural Networks for Efficient Pedestrian Detection," 2018, https://arxiv.org/abs/1805.08688.

[70] J. Nataprawira, Y. Gu, I. Goncharenko, and S. Kamijo, "Pedestrian detection using multispectral images and a deep neural network," *Sensors*, vol. 21, no. 7, p. 2536, 2021.

[71] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, "What can help pedestrian detection?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3127–3136, Honolulu, HI, USA, July 2017.

[72] G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection & segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4950–4959, Venice, Italy, October 2017.

[73] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3361–3369, Santiago, Chile, December 2015.

[74] J. Cao, Y. Pang, and X. Li, "Learning multilayer channel features for pedestrian detection," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3210–3220, 2017.

[75] T. Song, L. Sun, D. Xie, H. Sun, and S. Pu, "Small-scale Pedestrian Detection Based on Somatic Topology Localization and Temporal Feature Aggregation," 2018, https://arxiv.org/abs/1807.01438.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9240843, 16 pages
https://doi.org/10.1155/2022/9240843

Research Article

Relationship Discovery and Hierarchical Embedding for Web Service Quality Prediction

Hualong Bu,[1] **Jing Xia**,[2] **Qilin Wu**,[1] **and Liping Chen**[1]

[1]*School of Information Engineering, Chaohu University, Hefei 238000, China*
[2]*School of Mathematics and Statistics, Chaohu University, Hefei 238000, China*

Correspondence should be addressed to Qilin Wu; lingqiw@126.com

Received 20 April 2022; Accepted 24 September 2022; Published 5 October 2022

Academic Editor: Ripon Chakrabortty

Web Services Quality Prediction has become a popular research theme in Cloud Computing and the Internet of Things. Graph Convolutional Network (GCN)-based methods are more efficient by aggregating feature information from the local graph neighborhood. Despite the fact that these prior works have demonstrated better prediction performance, they are still challenged as follows: (1) first, the user-service bipartite graph is essentially a heterogeneous graph that contains four kinds of relationships. Previous GCN-based models have only focused on using some of these relationships. Therefore, how to fully mine and use the above relationships is critical to improving the prediction accuracy. (2) After the embedding is obtained from the GCNs, the commonly used similarity calculation methods for downstream prediction need to traverse the data one by one, which is time-consuming. To address these challenges, this work proposes a novel relationship discovery and hierarchical embedding method based on GCNs (named as RDHE), which designs a dual mechanism to represent services and users, respectively, designs a new community discovery method and a fast similarity calculation process, which can fully mine and utilize the relationships in the graph. The results of the experiment on the real data set show that this method greatly improved the accuracy of the web service quality prediction.

1. Introduction

More and more developers have paid attention to web services in the form of Web APIs [1]. By the end of March 22, 2022, ProgrammableWeb (the largest web service platform) has registered more than 24,417 APIs. Developers can access services through simple APIs, which provide a great convenience for development. As the number of services is increasing, users have only called a small number of services in reality, resulting in most service qualities being unknown in advance. Therefore, how to appropriately predict the quality of the web service becomes a serious issue [2].

Accompanied by the rising of deep learning technology, various deep web service quality prediction models have been continuously proposed [3–5]. Deep learning methods such as CNN and RNN can extract hidden information that cannot be treated by traditional matrix decomposition technology [6, 7] to improve the accuracy, but these methods do not take the topology information into account at all. In recent years, numerous graph convolutional networks (GCNs) have been proposed [8–12]. GCNs can integrate feature information from local graph neighborhood, which have been demonstrated to be powerful for graphical representation [13–15]. Neural Graph Collaborative Filtering (NGCF) [13] exploits the user-service graph structure and integrates the users and services interactions into the embedding process. PinSage [14] designs an efficient random walk method with graph convolutions to generate node embedding that incorporates both graph structure and node information. LDNM [15] transforms each service document into feature vectors by using LDA and Doc2vec, then applies Node2vec and MLP neural network for Web Service Classification.

Although GCNs have achieved great achievements in web service quality prediction, we argue that there are still suffering from the following challenges.

First, the user-service bipartite graph is essentially a heterogeneous graph. As shown in Figure 1, the bipartite user-service graph contains four types of relationships such as (1) user-user shared-invoking relationship; (2) user-user potential relationship; (3) user-server relationship; and (4) server-server relationship.

Taking the user node **u1** as an example, the first-order neighborhood of user **u1** contains the service nodes **s1** and **s2** which indicates the user-server relationship, and the second-order neighborhood contains the user node **u2** that has called the service **s2** before, which indicates the user-user shared-invoking relationship.

Taking the service node **s1** as an example, the first-order neighborhood of **s1** contains the user nodes **s1**, and the second-order neighborhood contains the service node **s2**. Node **s1** and node **s2** form the server-server relationship.

Furthermore, taking user **u1**, user **u5**, and user **u6** as an example, these users share characteristics (e.g., they may have common interests), but they have not revoked the same service in the past. In this way, users **u1**, **u5**, and **u6** form a user-user potential relationship with each other. Generally speaking, without prior domain knowledge, the user-user potential relationships are unknown in advance.

Previous GCN-based models have only focused on using one or more of these relationships. Therefore, how to fully mine the above relationships and make use of these relationships is critical to improve the prediction accuracy.

Second, when aggregating node features with the help of topology information, traditional methods must calculate the similarity. The existing similarity calculation research mainly focuses on how to improve the accuracy of downstream tasks, and it is necessary to traverse all samples one by one during traversal [16, 17]. Obviously, similarity calculation is a very time-consuming process when the number of samples is large. In order to reduce the computational complexity, a more efficient similarity calculation method needs to be adopted.

In addition, we notice that the web service quality prediction methods based on GCN have also faced the matrix sparsity problem to be solved. To address the aforementioned challenges simultaneously, this article proposes a novel relationship discovery and hierarchical embedding method (RDHE) based on GCNs for web service quality prediction, which can fully utilize four kinds of relationships to achieve better prediction accuracy. Given a user-service bipartite graph or QoS matrix, we first reconstruct the user-neighborhood graph and the service-neighborhood graph, respectively. Second, we propagate and aggregate the features of nodes from their local neighborhood by user embedding module and service embedding module, respectively. Third, we cluster the user embedding and service embedding with the help of clustering component. In addition, we conduct the communities using the community discovery module to instruct the downstream clustering. After these, we fuse the user-based and server-based predictions for the ultimate prediction. The experimental results show that our RDHE method can greatly improve the accuracy. The main contributions of this work are summarized as follows.

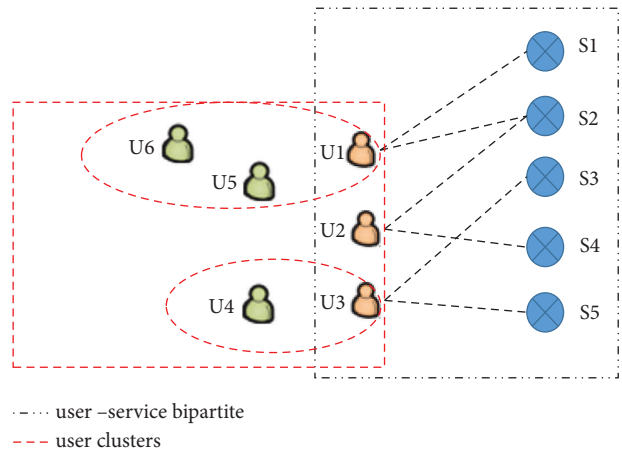


Figure 1: An illustration of the user and service relationships.

(1) A dual model to effectively learn neighborhood embedding from the user perspective and service perspective independently, which can lighten the damage caused by sparsity.

(2) Two community discovery methods are proposed to instruct the user- and service-based predictions to improve the accuracy of the final prediction, which can fully mine and use the above four kinds of relationship.

(3) A fast similarity calculation method to handle large-scale data, which can avoid the curse of dimensionality for downstream clustering tasks.

(4) A large number of experiments are carried out on the real WS-Dream data set. The experimental results show that compared with the classic methods, the prediction precision of our RDHE method improves a lot.

The remainder of this article is organized as follows: Section 2 introduces the related work for Web Service Quality prediction. Section 3 shows the whole framework. Section 4 explains in detail the implementation process of the RDHE algorithm. Section 5 presents the experimental results and analysis. We summarize the full text in the Conclusion section.

2. Related Work

2.1. Deep Learning Models for Embedding. With the development of cloud computing and the Internet of things, various web services are registered and invoked on the network, and the prediction of web service quality has become a popular research direction. Our work builds on some recent advances in graph neural network representation learning methods.

The concept of neural networks for graph data was first outlined by Gori et al. [18] and further elaborated by Scarselli et al. [19]. However, these initial methods required running expensive neural "message passing" algorithms to converge and were constrained by the scale of graph data, which was further improved by gated graph sequential neural networks [20] to graphs with <10,000 nodes, but the cost is still high.

A large number of approaches to "graph convolution" or graph convolutional networks (GCNs) have recently emerged, originating from the work of Bruna et al., which developed a graph convolution based on spectral graph theory [21]. Since then, many authors have proposed improvements, extensions, and approximations to these spectral convolutions [8–12, 22, 23], and achieved good results on tasks such as node classification and link prediction. These methods have consistently outperformed matrix factorization or random walk-based techniques (e.g., node2vec [24] and DeepWalk [25]), and their success has led to a surge of interest in applying GCN-based methods to applications such as recommender systems, of which Hamilton et al. [26] and Bronstein et al. [27] conducted a comprehensive survey of recent progress.

However, despite the success of GCN algorithms, the above works are mainly based on homogeneous graphs, ignoring the heterogeneous structure effects of graphs. Since the representation learning of heterogeneous graph neural network has the strong expressive ability and the characteristics of effectively combining node attribute features and structural information, it can not only solve the problems of network data such as data sparse [28], but also has achieved remarkable results in various downstream tasks, such as node classification [29], link prediction [30], node clustering [31, 32], and recommendation [33–35].

We found that the heterogeneity of the graph structure causes different node types to have different semantic features, and even if the node types are the same, there may be different community attributes (e.g., although they are all user nodes, their implied points of interest are different). Inspired by the community discovery algorithm [36], our work introduces community discovery (further discovering distinct communities of cohorts to guide downstream work) into graph neural network representation learning.

2.2. Similarity Calculation. Our method is also relevant to the field of similarity calculation. Similarity calculation indicators can evaluate the similarity between data, and provide an accurate and effective basis for data analysis. Previous studies on similarity calculation mainly focus on the following aspects:

(1) The design of basic measurement indicators, which can be divided into distance-based measurement methods and similarity coefficient-based measurement methods [37, 38]. Distance-based measurement methods mainly include Euclidean distance, Manhattan distance, etc., while similarity coefficient-based measurement methods mainly include Cosine similarity, Pearson correlation coefficient, etc. Among them, the most common methods are the application of Euclidean distance and Cosine similarity.

(2) Embedding similarity strategy design, this type of method mainly studies how to improve the guidance of similarity calculation for downstream tasks [16], such as ontology-based methods, semantic-based methods, and some other hybrid methods [17].

The above methods mainly consider the effectiveness or learning accuracy of the similarity measurement method itself, but due to the large amount of current data, the time performance of similarity calculation is insufficient. To this end, this article proposes a similarity calculation method based on a neural network. The main idea is to reduce the time required for similarity calculation as much as possible under the premise of ensuring accuracy.

3. The Whole Framework

We will briefly describe the framework of our proposed method RDHE, as shown in Figure 2. The entire framework consists of six components:

(i) User embedding module, which contains the reconstruction of the User Neighborhood component, Wide and Deep Embedding component, and GraphSAGE Embedding components.

(ii) Service embedding module, which contains similar components as user embedding module.

(iii) User-community discovery module, in which a new community discovery method is proposed to instruct the service-based clustering.

(iv) User clustering module, in which a clustering method is proposed for user clustering.

(v) Service clustering module, in which we get the service cluster and the user-cluster affiliation degree matrix.

(vi) At last, a fusion operation is used to fuse the user-based prediction and server-based prediction.

4. The Proposed Prediction Method

4.1. Reconstruction of Neighborhood Graph. Given a QoS matrix $Q^{m\times n}$, m and n are the number of users and services, respectively. $X^{(m+n)\times d}$ is the feature matrix and d is the dimension of features. The corresponding bipartite graph $G(U, S, E, X)$ can be simply derived from $Q^{m\times n}$. The bipartite graph $G(U, S, E, X)$ is a typical heterogeneous graph, the direct neighbors of users are the service nodes and the direct neighbors of services are the user nodes.

Assume that it is more efficient to learn different neighborhood embedding from the user perspective and service perspective independently, rather than recursively updating neighborhood embedding regardless of the node types. Based on this, we reconstruct a user-neighborhood graph and a service-neighborhood graph from the $G(U, S, E, X)$ or its corresponding QoS matrix directly. If user u_1 and user u_2 are connected to the same service, they become direct neighbors in the user-neighborhood graph. In other words, if the elements of the i-th column in the QoS matrix are nonzero, then corresponding users of service i become direct neighbors in the user-neighborhood graph.

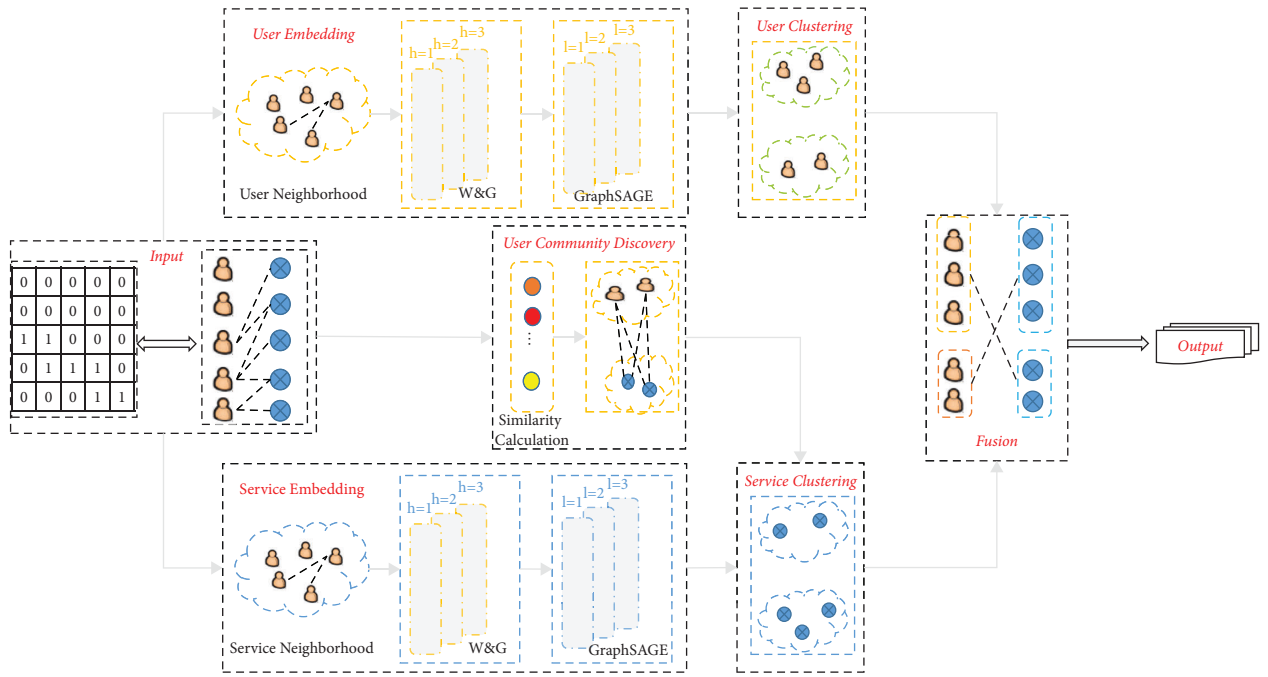


Figure 2: The framework of web service quality prediction. The input can be QoS matrix or corresponding bipartite graph. The output is the prediction of missing QoS values.

Likewise, if two services connect to the same user (the corresponding row's elements are nonzero), they become direct neighbors in the service-neighborhood graph. In this way, we can reconstruct the user-neighborhood graph $G(U, E', X_u)$, $X_u \in \mathbb{R}^{m\times d}$ and service-neighborhood graph $G'(S, E'', X_s)$, $X_s \in \mathbb{R}^{n\times d}$.

It should be noted that $G(U, E', X_u)$ contains two kinds of user relationship information, one is the user-user shared-invoking relationship which is constructed by the above method and is reflected in the topological structure of $G(U, E', X_s)$; the other is the user-user potential relationship which contains the latent community-sharing information, and we will discover it through the community discovery module (more details will be discussed in Section 4.3).

4.2. Sparsity and Large-Scale Handling. In general, the feature matrixes of X_u and X_s are sparse and high-rank, which will lead to the failure of methods which directly based on the feature matrixes. Here, we adopt the Wide & Deep [39] learning model to better represent sparse features by simultaneously training a linear LDA model and a neural network model.

Given a user-neighborhood graph $G(U, E', X_u)$ and its nodes feature matrix X_u, we can get the representation as,

$$E_u^w = WD_u X_u, \tag{1}$$

where $WD_u(\bullet)$ is a Wide & Deep learning module for users. Similarly, the service representation can be obtained by the Wide & Deep learning module as

$$E_s^w = WD_s X_s. \tag{2}$$

We then output E_u^w and E_s^w as inputs of GraphSAGE [40], which is a general framework that takes advantage of the node feature to efficiently embed large-scale data, and can generate embedding by sampling and aggregating features from a node's neighborhood as follows:

$$\begin{aligned} H_{\mathcal{N}_u}^t &= \text{aggregate}_u^t\left(\left\{H_i^{t-1}, i \in \mathcal{N}_u\right\}\right), \\ H_u^t &= \sigma\left(W_u^t \bullet \text{concat}\left(H_u^{t-1}, H_{\mathcal{N}_u}^t\right)\right), \\ H_{\mathcal{N}_s}^t &= \text{aggregate}_s^t\left(\left\{H_j^{t-1}, j \in \mathcal{N}_s\right\}\right), \\ H_s^t &= \sigma\left(W_s^t \bullet \text{concat}\left(H_s^{t-1}, H_{\mathcal{N}_s}^t\right)\right), \end{aligned} \tag{3}$$

where $\text{aggregate}_u^t(\bullet)$ and $\text{aggregate}_s^t(\bullet)$ are aggregating functions, σ is the activate function, W_u^t and W_s^t are the parameter matrixes which can be trained. $H_s^0 = E_s^w$ and $H_u^0 = E_u^w$ as original inputs.

Note that concat $(\bullet)$ is referring to the layer-aggregation mechanism. Since concatenation does not require additional parameters to learn, it has been shown quite simply and effectively for graph neural networks. In addition to concatenation, other strategies such as max pooling, can also be applied. Here, we adapt the concatenation for its simplicity.

4.3. Clustering with the Instruction of Community Discovery

4.3.1. Fast Similarity Calculation Process. Clustering divides a data set into different clusters according to a certain distance criterion so that the similarity of samples in the same cluster is as large as possible, and the difference of samples not in the same cluster is also as large as possible.

Traditionally, a clustering algorithm usually only needs a similarity calculation function $\mathrm{sim}(s_i, s_j)$ to get started. Traditional similar calculation process needs to traverse the embedding one by one when calculating $\mathrm{sim}(s_i, s_j)$, which are suitable for small-scale data sets, but the operating time will increase exponentially with the expansion of the dataset size [38]. In this work, a neural network-based similarity calculation process is designed for downstream similarity calculation and reduced the time complexity as shown in Figure 3.

In Figure 3, in order to improve the similarity calculation performance, a network model that can fit the similarity mapping relationship of the dataset is established. BP neural network has excellent nonlinear approximation, self-learning, self-adaptation, and generalization ability. According to the universal approximation theorem, if a feedforward neural network has a linear output layer and at least one hidden layer with an activation function, as long as there are enough hidden neurons, it can approximate any continuous function in a finite-dimensional space with arbitrary precision. In actual use, the computational time and space overhead must be considered, the exact representation is usually abandoned, and the nonlinear mapping relationship between the dataset and the label set is approximated by finding suitable parameters on the basis of the approximate representation. This provides a theoretical basis for using BP neural network for fast similarity calculation.

After the similarity calculation formula is selected, the numerical value of the data and the similarity between the data can constitute a definite multi-input and multi-output nonlinear mapping relationship, so the BP neural network can be used to fit this mapping relationship. In order to reduce the time of traversal calculation, the model is trained based on the similarity of some samples, and the accuracy is exchanged for speed, and the nonlinear similarity mapping relationship of the complete data set is represented within the allowable error range, so as to calculate the approximate similarity of the complete data set.

We give a brief complexity analysis here. Set $X \in R^{n\times m}$. For tradition calculating method, the total time complexity is $O(mn^2)$. The time complexity of the neural network method can be divided into precise calculation part and network part. In the precise calculation part, let b be the sampling ratio, $0 < b < 0.5$, the precise calculation complexity is $O(b^2mn^2)$. Assuming the number of the inputted neurons per hidden layer is m, the hidden layer number is t, and the number of output neurons is 1, the time complexity of the neural network method is $O((1 - \mathrm{b})mnt)$. Generally, $t \ll n$, so the neural network method is more efficient than the traversal calculation method.

4.3.2. User Clustering. We note the cluster center vector as the mean vector of the feature vectors of all the users in the cluster, and calculate the similarity between the feature vector of the user u in cluster i and the rating vector of cluster i as the user u's interest in the cluster i.

The similarity calculation function is the widely used Pearson Correlation Coefficient (PCC). We use PCC to measure the similarities between vector r_i and r_i, and get the user clusters UC on the user embedding H_u^t. We use the function $f(x) = (x + 1)/2$ to project the PCC into range [0, 1]. The PCC is defined as:

$$\mathrm{sim}(i, j) = \frac{\sum_{p\in P}\left(x_{ip} - \overline{x}_i\right)\left(x_{jp} - \overline{x}_j\right)}{\sqrt{\sum_{p\in P}\left(x_{ip} - \overline{x}_i\right)^2\left(x_{jp} - \overline{x}_j\right)^2}}, \tag{4}$$

where P is the interaction of the two vectors.

Based on the assumption that users with more social relationships are more representative, we can derive the weighted cluster center vectors.

$$WC_i = \frac{\sum_{g\in UC(i)}\left|relation_i(g)\right|r_{u_g}}{\sum_{g\in UC(i)}|UC(i)|}, \tag{5}$$

where $C(i)$ represents the set of users in cluster i, $\mathrm{relation}_i(g)$ represents the set of users who have a direct relationship with user u_g in cluster i. r_{u_g} is the feature vector of user u_g.

Finally, we can obtain the user clusters UC and the rating matrix UCR by formula (4) and (5).

4.3.3. Service Clustering with the Instruction of Community Discovery. Finding and utilizing community information on the original input QoS matrix without prior knowledge can be connected to the concept of community discovery. A fuzzy clustering algorithm based on interest preference is proposed here. The algorithm uses the user's behavior records and the clusters to which the services belong to find a set of users similar to the target user's interest preference.

Given the original input QoS matrix $Q^{m\times n}$, the users' ratings on services form a $m \times n$ rating matrix $R = [r_{ij}]$, where r_{ij} is the rating of user u_i on service s_i, r_i is the rating vector of user u_i. We note the community rating vector as the mean vector of the rating vectors of all the users in the community.

First, we use the PCC to calculate the similarity of the service embedding H_s^t to get the service clusters SC.

Next, we derive the distribution of clusters SC which have been rated by u_i can be described as:

$$\mathrm{Dist}(i) = \left(\frac{\left|S_i^{SC_1}\right|}{|S_i|}, \frac{\left|S_i^{SC_2}\right|}{|S_i|}, \ldots, \frac{\left|S_i^{SC_p}\right|}{|S_i|}\right), \tag{6}$$

where $S_i$represents the set of services rated by user u_i, and $S_i^{SC_k}$ represents the set of services rated by user $u_i$in cluster C_k.

Next, due to the sparse data of the user-service rating matrix, the user category preference vector also has a sparseness problem, we define a more effective similarity function than traditional Minkowski distance as follows:

$$\mathrm{sim}\left(Dist(i), g_j\right) = \frac{1}{\tau_i}\sum_{k=1}^{p}\tau_{ik}\left(Dist(i)_k, g_{j,k}\right), \tag{7}$$

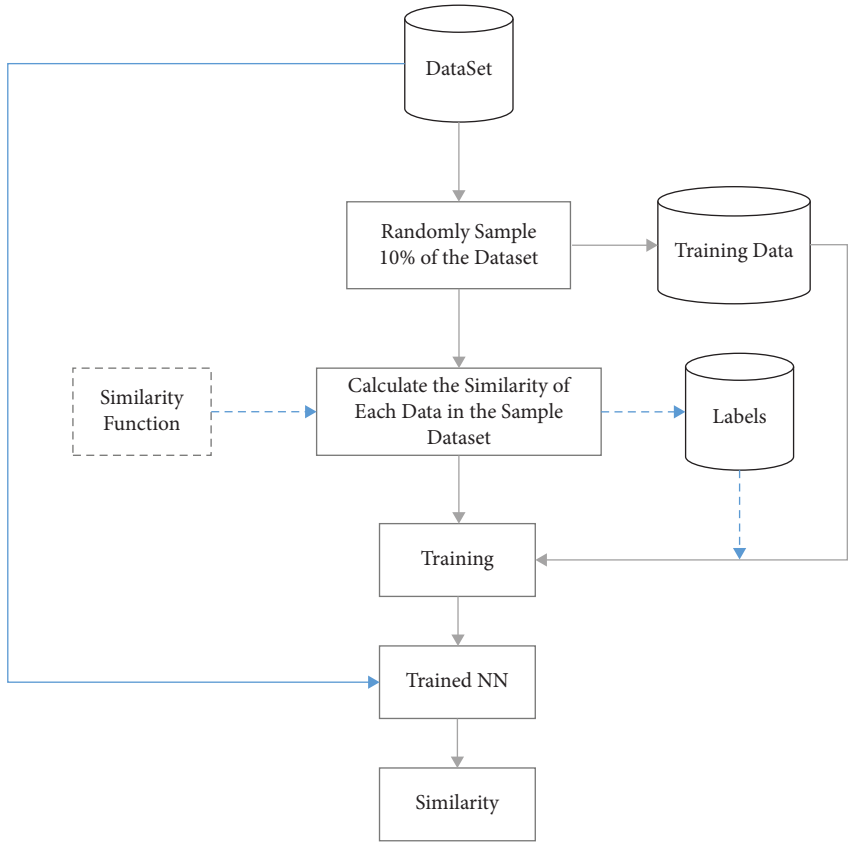


Figure 3: Fast similarity calculation.

where $\tau_{ik} = \begin{cases} 1, Dist(i)_k \neq 0 \\ 0, Dist(i)_k = 0 \end{cases}$, $\tau_i = \sum_{k=1}^{p} \tau_{ik}$. l is the number of clusters, g_j represents a certain cluster center vector, $j = 1, 2, \ldots, l$. $g_{j,k}$ is the k-th element in g_j.

Then we derive the objective function:

$$\text{Objective}(USA, G) = \sum_{i=1}^{m} \sum_{j=1}^{l} u_{ij}^2 \text{sim}\left(Dist(i), g_j\right), \tag{8}$$

where $u_{ij} \in [0, 1]$ represents the affiliation degree of user u_i to cluster C_j and $\sum_{k=1}^{l} u_{ij} = 1$. $USA = u_{ij}^{m \times l}$ represents the user-cluster affiliation degree matrix. $G = (g_1, g_2, \ldots, g_l)^T$ represents the center cluster matrix.

Finally, we update the values of the user-cluster affiliation matrix USA and cluster center matrix G through iterative operation, and gradually reduce the error of the objective function to the pre-set convergence threshold and terminate the iteration.

4.4. Joint Prediction. In order to reduce the influence of the redundant neighbors on the prediction, we predict the missing QoS values $P_u(i, j)$ of user j as follows:

$$P_u(i, j) = \frac{\sum_{i=1}^{\text{top}-k} \left(qos_{u_i} \times UCR_{it}\right)}{\text{top} - k}, \tag{9}$$

where u_i is the i-th user in the cluster UC_t which user u_j belongs to, qos_{u_i} represents the QoS value of user u_i, UCR_{it} means the rating score of user u_i to cluster UC_t, and top $-$ k represents the number of how many neighbors to be selected.

The formula for predicting missing values $P_s(i, j)$ of service j based on the service clustering result is as follows:

$$P_s(i, j) = \frac{\sum_{i=1}^{top-k} \left(qos_{u_i} \times US_{it}\right)}{\text{top} - k}, \tag{10}$$

where u_i is the i-th affiliated user of the server cluster SC_t which user u_j affiliated to, qos_{u_i} represents the QoS value of user u_i, US_{it} means the affiliation degree of user u_i to cluster SC_t, and top $-$ k represents the number of how many neighbors to be selected.

To fully consider the similarity of similar users and similar services, this article combines formulas (9) and (10) to propose a hybrid prediction method as follows:

$$P_{u,s}(i, j) = \lambda \times P_u(i, j) + (1 - \lambda) \times P_s(i, j). \tag{11}$$

The parameter $\lambda \in [0, 1]$ indicates that the predicted value depends on the proportion of similar users and similar services. When $\lambda = 1$, only the user-based method is used for prediction; when $\lambda = 0$, only the service-based method is used for prediction. If user u does not call any service that has been used or service s has not been called by any user, the average of all service QoS values will be used for prediction.

4.5. Model Training. To optimize the model parameters of RDHE, we need to optimize an objective function. In the training process, we denote S^+ as positive samples and denote S^- as negative samples, respectively. Given user $u, y_{u,s}$ is the ground truth label, if user u has invoked the service s, then $y_{u,s} = 1$, otherwise $y_{u,s} = 0$, $\widehat{y}_{u,s}$ is the predicted label for service s. The loss function is derived as:

$$\% - \sum_{u} \sum_{i \in S^+ \cup S^-} \left\{ y_{u,s} \log P\left(\widehat{y}_{u,s} = 1\right) + \left(1 - y_{u,s}\right) \log P\left(\widehat{y}_{u,s} = 0\right) \right\}. \tag{12}$$

Our model contains service embedding E_s^F and user embedding E_u^F, they are hierarchical trained and learned together. Since the raw features are very large and highly sparse for large-scale graphs, we do not use one-hot vectors to represent the nodes. By embedding high-dimensional sparse features into a low-dimensional latent space, the model can be easy to train. It is worth noting that, in terms of model size, the majority of the RDHE model parameters come from the embedding. We adopted the Cluster-GCN Mini-batch Trick [41] and Adam [42] optimization for GCNs.

Overfitting is another long-term problem for optimizing deep neural network models. The dropout strategy [43] is a common method to solve the overfitting of the neural network, which has the characteristics of simplicity and good effect. Here, we employ node and feature dropout strategies to prevent RDHE from overfitting. When the parameters, only part of them will be updated. Moreover, as a dropout is disabled during testing, the whole network is used for prediction.

5. Experiment

In this work, we performed a lot of experiments on real-world datasets to evaluate our RDHE method. We hope to answer the following questions:

(i) How does RDHE perform compared to the state-of-the-art methods?

(ii) How do different hyper-parameter settings affect RDHE?

5.1. Experimental Data and Evaluation Indicators. In our experiments, in order to evaluate the effectiveness of the prediction method proposed in this article, we use the WS-Dream dataset [44]. The two matrices in this dataset contain 1974675 response time (RT) and throughput (TP) data generated by 339 users calling 5825 services. There is a QoS record generated by invoking between each user and each Web service. The real recorded values of RT are between [0, 1].

To evaluate the effectiveness of service quality prediction, this work uses the mean absolute error (MAE) and the root mean square error (RMSE) indicators for test, which are defined as follows.

$$\begin{aligned} \text{MAE} &= \frac{\sum_{u,s} |r(u,s) - p(u,s)|}{N}, \\ \text{RMSE} &= \sqrt{\frac{\sum_{u,s} (r(u,s) - p(u,s))^2}{N}}, \end{aligned} \tag{13}$$

where $r(u,s)$ and $p(u,s)$ represent the actual QoS and predicted QoS values, respectively, and N represents the number of predictions. It can be seen from the formula that the smaller the values of MAE and RMSE, the better the prediction performance, and the RMSE is more sensitive to the prediction error.

5.2. Baselines. We compare our proposed RDHE method with 10 state-of-the-art methods, which is described as follows:

(1) LDA [45]: LDA is an unsupervised topic modeling algorithm, which has been widely used in recommender systems. Here, each service is associated with a distribution on the topic of users, each user group is a distribution of users, and we use LDA to discover hidden relationships between users and services in the prediction task.

(2) Node2Vec [24]: Node2Vec is an improved version of DeepWalk that takes into account structure and homogeneity. In this method, we use the Node2vec model to maintain node neighbor information from the user- and service-neighborhood graphs for downstream tasks.

(3) GCN [8]: GCN is a classic semi-supervised model for extracting graph features. In this article, we treat both the services and users as nodes, then use the GCN to obtain the embedded representation of the graph.

(4) Wide & Deep [39]: in this method, we only use the Wide & Deep method to obtain user and service embedding for downstream similarity calculation and prediction tasks.

(5) GraphSAGE [40]: in this method, we only use the GraphSAGE module to obtain user and service feature representations for downstream tasks.

(6) NGCF [13]: in this method, we use the NGCF module to exploit the user-service graph structure by propagating high-order connectivity embedding on it; next the prediction task is imposed on it.

(7) PinSage [14]: in this method, we use a PinSage module to combine to generate embedding of nodes that incorporate both graph structure as well as node feature information. Then we perform similarity calculation and prediction tasks on it.

(8) KGAT [34]: in this method, we use a KGAT module to propagate the embedding by aggregating information from the constructed knowledge graph for prediction.

(9) GraphRec [12]: in this method, we use a GraphRec module to capture interactions and heterogeneous strengths in the user-service graph for prediction.

(10) NIA-GCN [35]: in this method, we adopt the NIA-GCN module to capture relationships between pairs of neighbors at each GCN layer for prediction.

LDNM is a fusion method of LDA, Doc2vec, and Node2vec, which have been adopted, respectively, thus we will not compare it.

5.3. Parameter Setting. We implement our method on the basis of Pytorch2 and use the Adam optimizer for all neural network-based methods. All the baselines and our algorithm are fully trained with up to 500 epochs, and the number of negative samples is 20. Moreover, we use validation set data for parameter tuning for all the methods.

In the real world, most of the users have only called a few services, therefore, in this work, to make the experiments more realistic, we randomly delete a certain number of QoS values from the initial RT and TP matrices to generate low-density matrices. For example, a matrix density of 5% means that we randomly select 5% of the QoS values to predict the remaining 95% of the QoS values. The removed QoS values are used as expected values to study prediction accuracy. Each training set data contain 10% of the validation set data; for example, 50% of the training set size means that 40% of the data are used for training and the remaining 10% of data are for validation.

The default values of the parameters and hyper-parameters of the baselines are shown in Table 1. If there are multiple datasets and the parameters are not specified, then the default parameters are the same. Specifically, we set the number of topics to 20 for all LDA-contained methods, where each latent topic corresponds to a specific service cluster. For our method, the Wide & Deep and GraphSAGE parameters are the same as in Table 1. We set the hidden layer the same as the embedding size and the activation function as ReLU. We used four hidden layers for the NN components of the Similarity Calculation and Prediction Module and its learning rate = 0.03, dropout = 0.5, and optimizable parameter $\delta = 0.6$. The effect of different values of λ, top-k, and other hyper-parameters will be shown in chapter 5.4.

5.4. Experiment Result. We first compare the prediction accuracy of different methods and then compare the effect of different parameters in RDHE, all the results and details will be discussed below.

5.4.1. Prediction Accuracy Comparison. In the experiment, the matrix density is set to 0.1–0.3 on the WS-Dream dataset (divided into RT and TP sub-datasets), and gradually increases at an interval of 0.05. We set $k = 3$ and $\lambda = 0.4$ to obtain the MAE and RMSE results. The experimental results are compared in Tables 2 and 3.

Tables 2 and 3 compare the prediction accuracy of 10 state-of-the-art methods with RDHD. It can be seen that our RDHE method has obtained better experimental results with both MAE and RMSE indicators, compared to all other baseline methods. Take the result of the MAE indicator as an example, the RDHE method is better than the four classic graph algorithms (Node2vec, GCN, Wide & Deep, and GraphSAGE) with 18.84%, 21.50%, 15.47%, and 12.50% achievements, respectively; it also achieves 11.11%, 9.19%, 9.19%, 5.62%, and 3.45% improvement than NGCF, PinSage, KGAT, GraphRec, and NIA-GCN, respectively. Similarly, take the result of the RMSE indicator in Table 3 as an example, the RDHE method achieved 14.48%, 16.85%, 14.19%, 13.74%, 12.33%, 8.51%, 7.98%, 11.21%, and 8.04% accuracy improvement than the nine GCNs methods based on the RMSE indicators.

Tables 2 and 3 also show the prediction accuracy changes of the different methods as the matrix density increases. As the matrix density increases, each method achieves better results than at lower densities, and the RDHE method improves even more.

5.4.2. The Influence of λ for Prediction Accuracy. As shown in (11), the parameter λ is a weight parameter, which determines the degree of dependence of the results based on the user's similarity and service's similarity. For example, if $\lambda = 1$, only the user-based method is used for prediction, and if $\lambda = 0$, only the service-based method is used for prediction. In order to study the influence of λ on the prediction results, the matrix density is set to 0.1, the number of similar users and the number of similar services is 6, and the λ ranges from 0.1 to 0.9 with an interval of 0.05. The experimental results are shown in Figure 4.

Among them, Figures 4(a) and 4(b) are the experimental results of the response time attribute, and Figures 4(c) and 4(d) are the experimental results of the throughput attribute. The experimental results in Figure 4 show that λ is one of the decisive factors for RDHE prediction accuracy. As shown in Figures 4(a) and 4(c), as the value of λ increases, the prediction accuracy reaches the maximum value when $\lambda = 0.3$. Meanwhile, as shown in Figures 4(b) and 4(d), the prediction accuracy reaches the maximum value when $\lambda = 0.4$. This shows that a suitable λ value can achieve better prediction accuracy. At the same time, the results in Figure 4 show that when the datasets are different, the optimal value of λ is also different.

5.4.3. The Influence of Top-k for Prediction Accuracy. The parameter k can determine how many similar users or similar services of the selected target are to be used. Here, in order to study the influence of top k-th values on prediction accuracy, we set the matrix density to 0.05 and $\lambda = 0.4$, and set the Top-k value from 1 to 10 and gradually increase at intervals of 1, the results are shown in Figure 5. Among them, Figures 5(a) and 5(b) are the experimental results in the response time attribute, and Figures 5(c) and 5(d) are the experimental results in the throughput attribute. It can be seen from Figures 5(a) and 5(d) that with increasing the top-

Table 1: Default main hyper parameters setting.

Modules	Main hyper parameters	Value
LDA	Number of topics in LDA	20
	Dirichlet parameter	0.1
Node2vec	Length of Node2vec walk	80
	Number of Node2vec walks	10
	Window size of Node2vec walk	10
GCN	Number of layers	2
	Size of layer convolution layer	64
Wide & deep	Number of layers	2
	Dimensions	128
GraphSAGE	Neighborhood deep neighborhood sample sizes	2 25
NGCF	Embedding propagation layers	3
	Depth of the NGCF	3
	Top-k	20
PinSage	Top-k	2
	Size of hidden dimension size	512
KGAT	Top-k	20
	Tower structure of hidden layer size	512/256/128/64
GraphRec	Size of the hidden layer	128
	Number of three hidden layers	3
NIA-GCN	Size of first layer/second layer	40/2
	Dimension of each layer	64

Table 2: Overall performance comparison of RT on average.

Method	MAE/matrix density						RMSE/matrix density					
	0.10	0.15	0.20	0.25	0.30	Average	0.10	0.15	0.20	0.25	0.30	Average
LDA	0.49	0.46	0.45	0.43	0.42	0.4500	1.48	1.45	1.44	1.43	1.41	1.4420
Node2vec	0.44	0.43	0.41	0.40	0.39	0.4140	1.43	1.41	1.4	1.38	1.37	1.3980
GCN	0.45	0.43	0.43	0.42	0.41	0.4280	1.47	1.45	1.43	1.41	1.39	1.4300
Wide & deep	0.42	0.41	0.39	0.38	0.37	0.3975	1.39	1.36	1.32	1.30	1.28	1.3300
GraphSAGE	0.41	0.4	0.38	0.37	0.36	0.3840	1.38	1.37	1.35	1.34	1.32	1.3520
NGCF	0.4	0.38	0.37	0.38	0.36	0.3780	1.37	1.36	1.35	1.34	1.33	1.3500
PinSage	0.4	0.38	0.36	0.36	0.35	0.3700	1.35	1.34	1.32	1.31	1.30	1.3240
KGAT	0.39	0.38	0.37	0.36	0.35	0.3700	1.31	1.29	1.27	1.26	1.25	1.2760
GraphRec	0.38	0.36	0.35	0.35	0.34	0.3560	1.3	1.27	1.25	1.23	1.22	1.2540
NIA-GCN	0.39	0.36	0.34	0.33	0.32	0.3480	1.29	1.26	1.23	1.22	1.21	1.2420
RDHE	**0.37**	**0.35**	**0.34**	**0.32**	**0.3**	**0.3360**	**1.24**	**1.22**	**1.21**	**1.20**	**1.20**	**1.2140**

Table 3: Overall performance comparison of TP on average.

Method	MAE/matrix density						RMSE/matrix density					
	0.10	0.15	0.20	0.25	0.30	Average	0.10	0.15	0.20	0.25	0.30	Average
LDA	16.29	15.10	14.02	13.97	13.73	14.6220	58.25	57.2	56.56	55.1	54.55	56.3320
Node2vec	14.85	13.76	13.35	13.19	12.82	13.5940	56.85	55.98	54.15	53.29	51.18	54.2900
GCN	15.25	14.30	14.01	13.83	13.55	14.1880	57.93	57.1	55.97	54.35	53.83	55.8360
Wide & deep	14.98	14.21	14.02	13.87	13.85	14.1860	56.09	55.12	54.4	52.97	51.95	54.1060
GraphSAGE	14.76	14.10	13.66	13.11	12.97	13.7200	55.57	54.04	53.52	52.81	53.19	53.8260
NGCF	14.85	14.20	13.86	13.31	12.85	13.8140	55.06	54.02	53.21	51.56	50.94	52.9580
PinSage	14.69	14.15	13.78	13.23	12.95	13.7600	52.87	51.83	50.3	49.78	48.96	50.7480
KGAT	14.76	14.10	13.66	13.11	12.97	13.7200	51.87	51.33	50.25	49.98	48.85	50.4560
GraphRec	14.18	13.53	12.95	12.86	12.75	13.2540	54.23	53.6	52.15	51.42	50.07	52.2940
NIA-GCN	13.58	13.22	13.05	12.99	12.87	13.1420	52.23	51.73	50.05	49.68	48.77	50.4920
RDHE	**12.58**	**12.49**	**12.45**	**12.40**	**12.17**	**12.4180**	**48.26**	**47.41**	**46.16**	**45.32**	**45.01**	**46.4300**

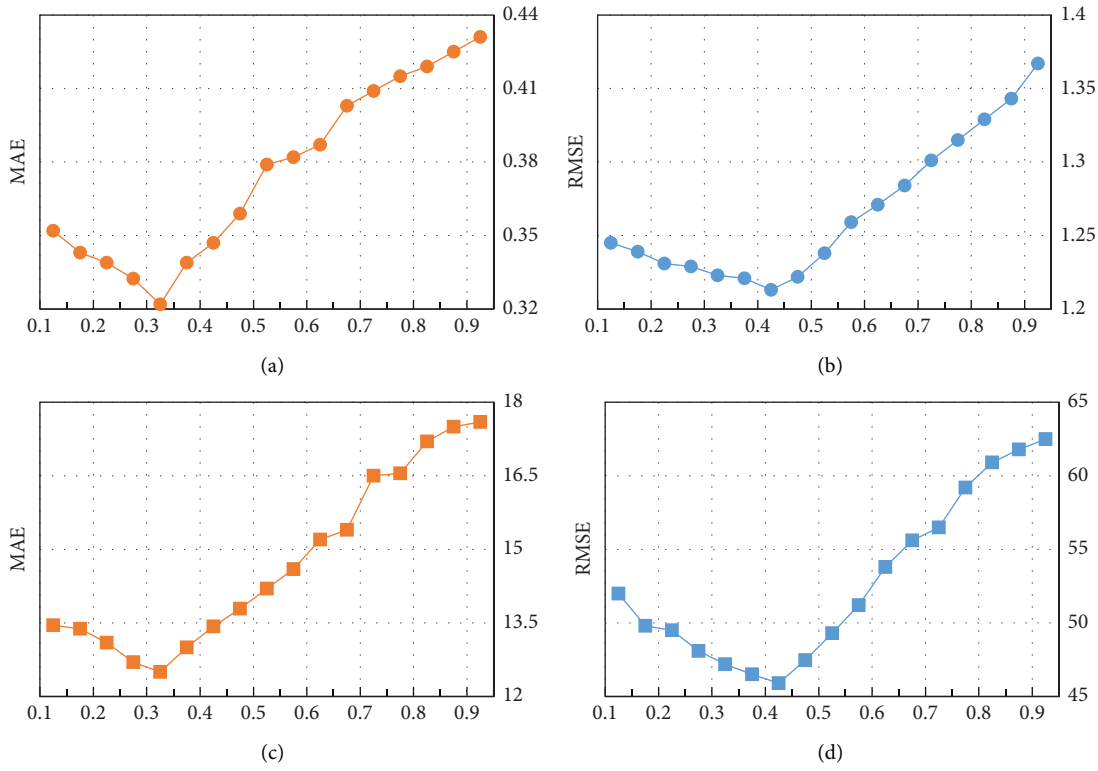


Figure 4: The influence of parameter λ. (a) RT attribute. (b) RT attribute. (c) TP attribute. (d) TP attribute.

k value, the prediction accuracy first increases, then decreases, and reaches the maximum when k is 4. This shows that the number of similar users or similar services is not always beneficial to improving the prediction accuracy, too few or too many similar users and similar services can both reduce the prediction accuracy.

5.4.4. The Influence of Matrix Density. The matrix density represents what proportion of the QoS matrix is used as training data. Figure 6 shows the effects of matrix density on two QoS attributes, response time, and throughput, respectively. Here, we set the density of the matrix from 0.05 to 0.25, increasing in 0.05 intervals, $\lambda = 0.4$, the number of similar users, and the top-k is 4.

We can see from Figure 6 that, as the density of the matrix increases, the service prediction accuracy of the RDHE method gradually improves. This is because the larger the matrix density is, the more users have invoked more services, and the generated data set contains more user information and service information, from which more useful information can be mined to improve prediction accuracy.

5.4.5. The Influence of Hyperparameters. Our method includes three neural networks (wide & Deep, GraphSAGE, and MLP for clustering), among which the default values of the parameters have been depicted in Table 1, more discussion can be referred in [39, 40]. Here, we mainly discuss the key parameters of NN, which are crucial for clustering, including the number of hidden layers, learning rate, and dropout.

(1) Impact of Hidden Layers. In theory, the multi-hidden layer network structure can learn patterns from the original data, and express the data more abstractly through layer-by-layer feature extraction. Therefore, the more layers, the stronger the expressive ability of the data and the stronger the prediction ability. But this improvement is not without an upper limit, and when the number of hidden layers is too high, the accuracy may drop.

To study the number of impact of the hidden layers, we vary the number of hidden layers from 1 to 8, and the dimension of the hidden layer is set to 64. As shown in Figure 7, in both the RT and TP attributes, when the number of hidden layer increases, the value of MAE also increases. When the number of hidden layers is 4, the optimal prediction effect can be achieved. Therefore, we choose the number of hidden layers as 4 in our method.

(2) Impact of Learning Rate. The learning rate is one of the key hyper-parameters for training neural networks. When the learning rate is relatively small, the loss curve converges very slowly, but the amplitude of the loss swing is relatively small because the amplitude of the weight update amplitude is small. Otherwise, when the learning rate is relatively large, the loss curve converges quickly. Therefore, it is very important to choose the learning rate carefully for our method. In order to find the optimal value of the learning rate, we set the learning rate as 0.01, 0.02, 0.03, 0.04, 0.05, and 0.1. Figure 8 shows the performance change of RDHE.

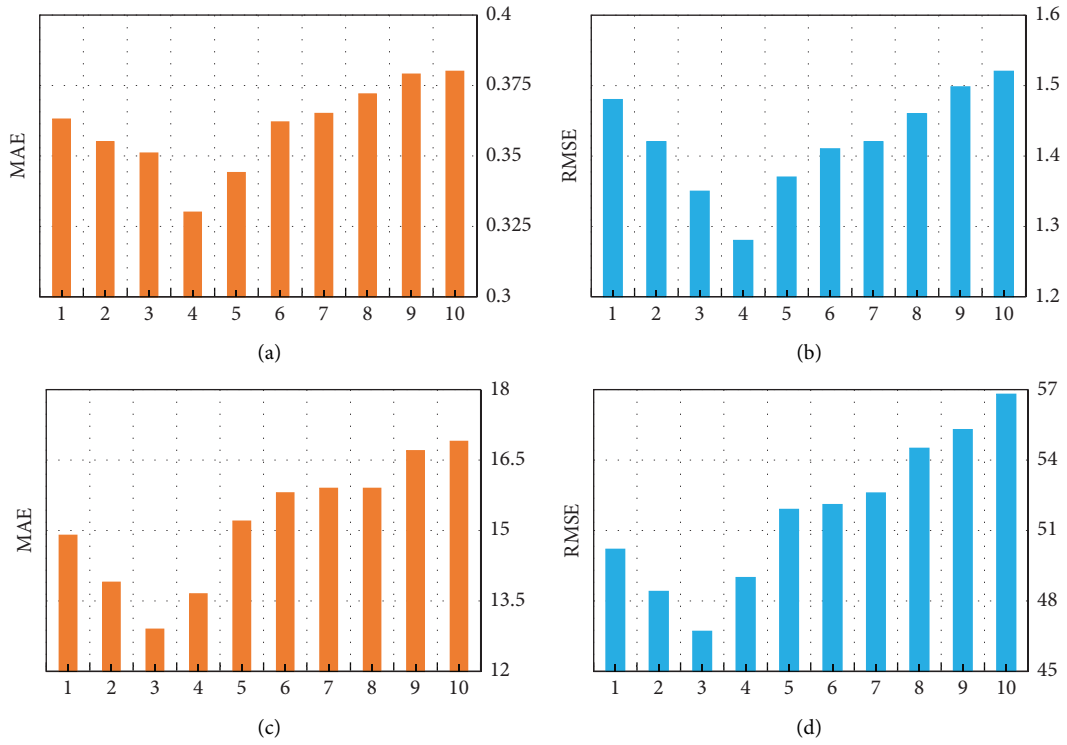


Figure 5: The influence of parameter top-*k*. (a) Top-k on RT attribute. (b) Top-k on RT attribute. (c) Top-k on TP attribute. (d) Top-k on TP attribute.

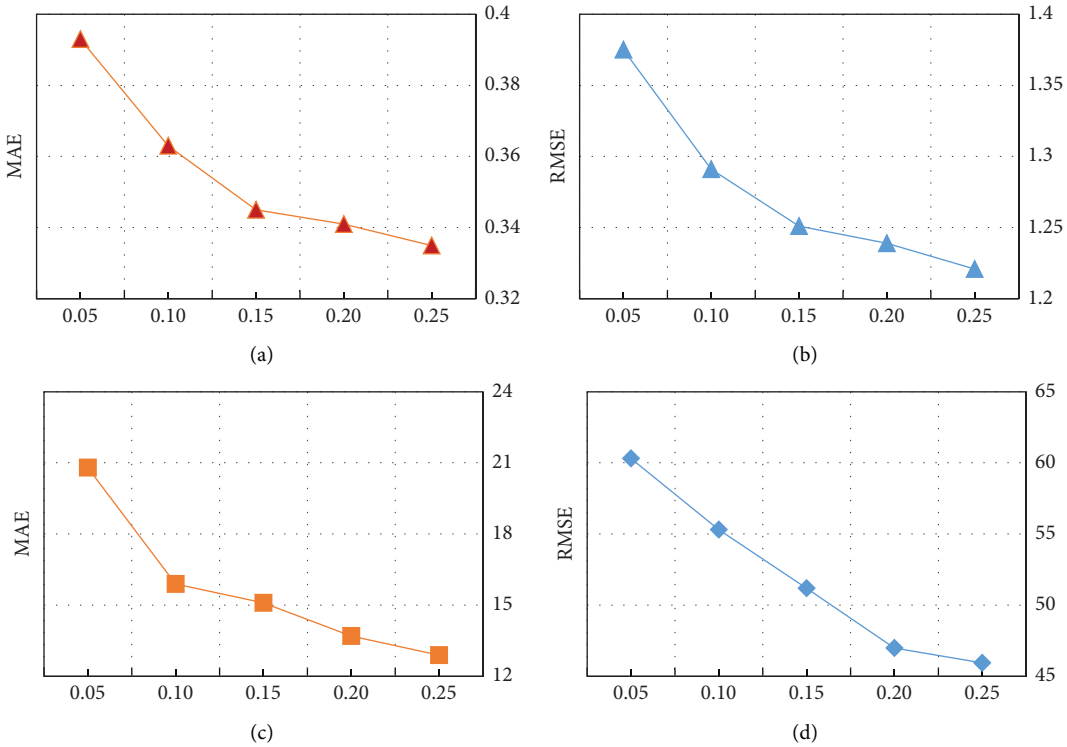


Figure 6: The effect of parameter matrix density. (a) Matrix density on RT. (b) Matrix density on RT. (c) Matrix density on TP. (d) Matrix density on TP.

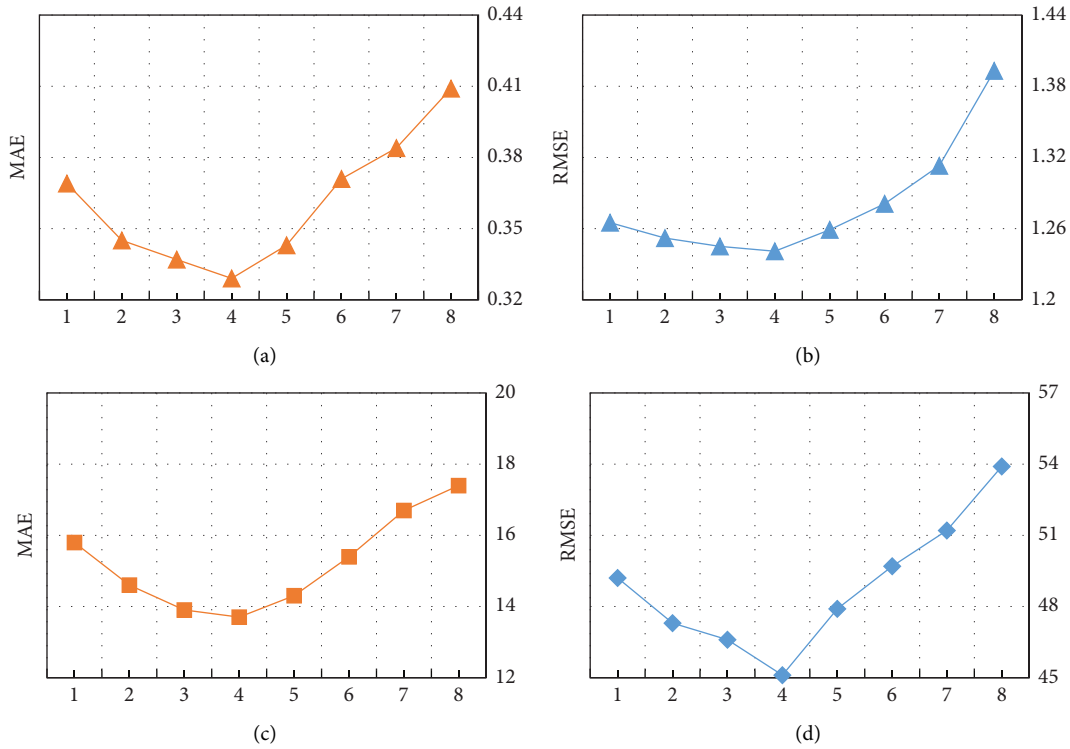


Figure 7: Comparison of the hidden layer numbers. (a) Hidden layer number on RT. (b) Hidden layer number on RT. (c) Hidden layer number on TP. (d) Hidden layer number on TP.

From Figures 8(a) and 8(b), we can observe that the learning rate improves the performance when the value range from 0.01 to 0.03 on the RT attribute, and the improvement of the classification effect in the TP attribute does not stop until the learning rate is set to 0.03, this is because a learning rate that is too small can cause the process to get stuck and the method cannot approach the local minima. However, as the value of the learning rate varies from 0.03 to 0.1 on the RT attribute and the TP attribute, the classification performance gradually decreased; the possible cause of this phenomenon is that a learning rate too large will cause the model to converge too quickly to a suboptimal solution. Therefore, in our experiments, the default learning rate is set to 0.03.

(3) Comparative Results of Node and Feature Dropout. Although GCN-based learning models have strong representation ability, they are usually affected by overfitting. The dropout strategy is a common method for solving the overfitting of the neural network, which has the characteristics of simplicity and good effect. Here, we employ node and feature dropout strategies to prevent RDHE from overfitting. The feature dropout strategy randomly removes the output information with a probability of p_n. As a result, only partial information contributes to the refined representations in the l-th propagation layer. We also run a node dropout strategy to randomly block a particular node and discard all its output information with a probability p_f.

Figure 9 shows the effect of the node dropout ratio and the feature dropout on different attributes of the WS-Dream dataset. From Figure 9, we can find that the node dropout strategy gets better accuracy than feature dropout both on (a), (c), and (d). Taking (a) Dropout Ratio on RT as an example, setting the radio as 0.4 will lead to the highest MAE result of 0.329, which is better than that of the feature dropout result of 0.362. One possible reason is that dropping some features of user or service nodes can make the representation more robust; therefore, node dropout is more effective than feature dropout, which means that the node dropout strategy can be a solution to graph neural network overfitting.

5.4.6. Experimental Analysis of Fast Similarity Calculation Method. In this work, a neural network-based similarity calculation process is designed for downstream similarity calculation and reduced the time complexity. In order to verify the effectiveness of our method, we designed two comparison experiments on TP, (1) FSCM, which had fast similarity calculation component and (2) Non-FSCM, which traversed the embedding one by one directly when calculating $\mathrm{sim}(s_i, s_j)$, the experiment selected five models: LDA, GraphSage, GCN, NGCF, and RDHE for comparison, the unit of Runtime is hours, and the runtime required for the comparison experiment is shown in Table 4.

It can be seen from Table 4 that the addition of the fast similarity calculation component is 4 hours (25.48% less than Non-FSCM) less than the Non-FSCM method on RT and 3.9 hours (24.69% less than Non-FSCM) on TP. It is

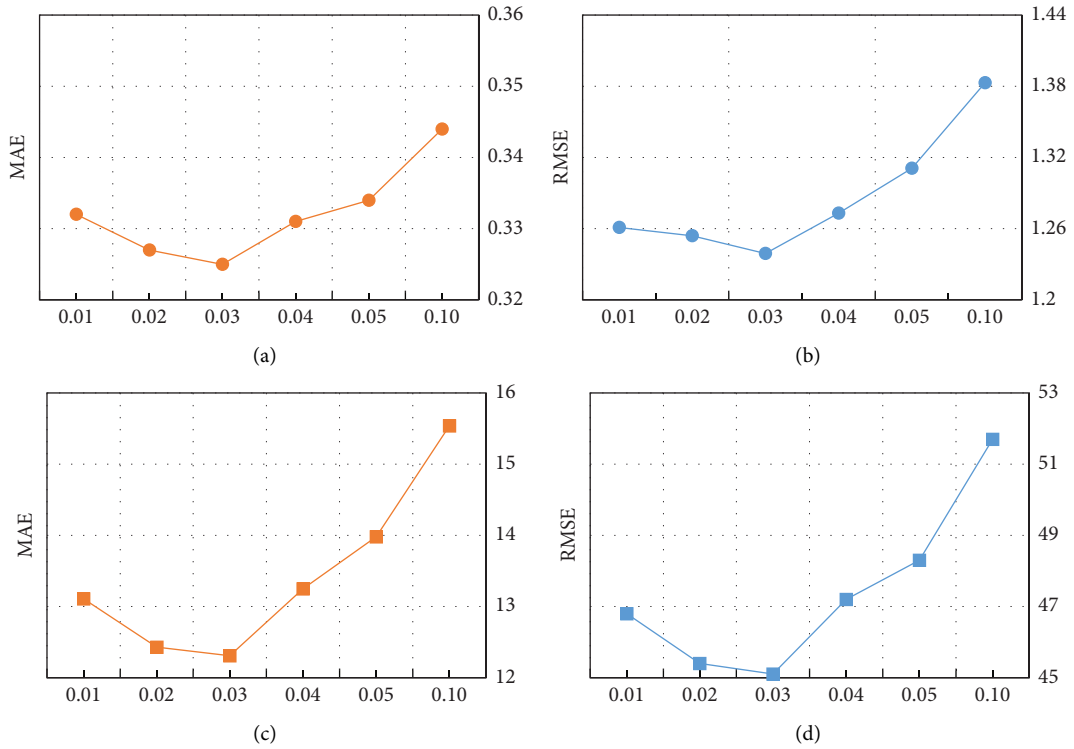


Figure 8: Comparison of the learning rate. (a) Learning rate on RT. (b) Learning rate on RT. (c) Learning rate on TP. (d) Learning rate on TP.

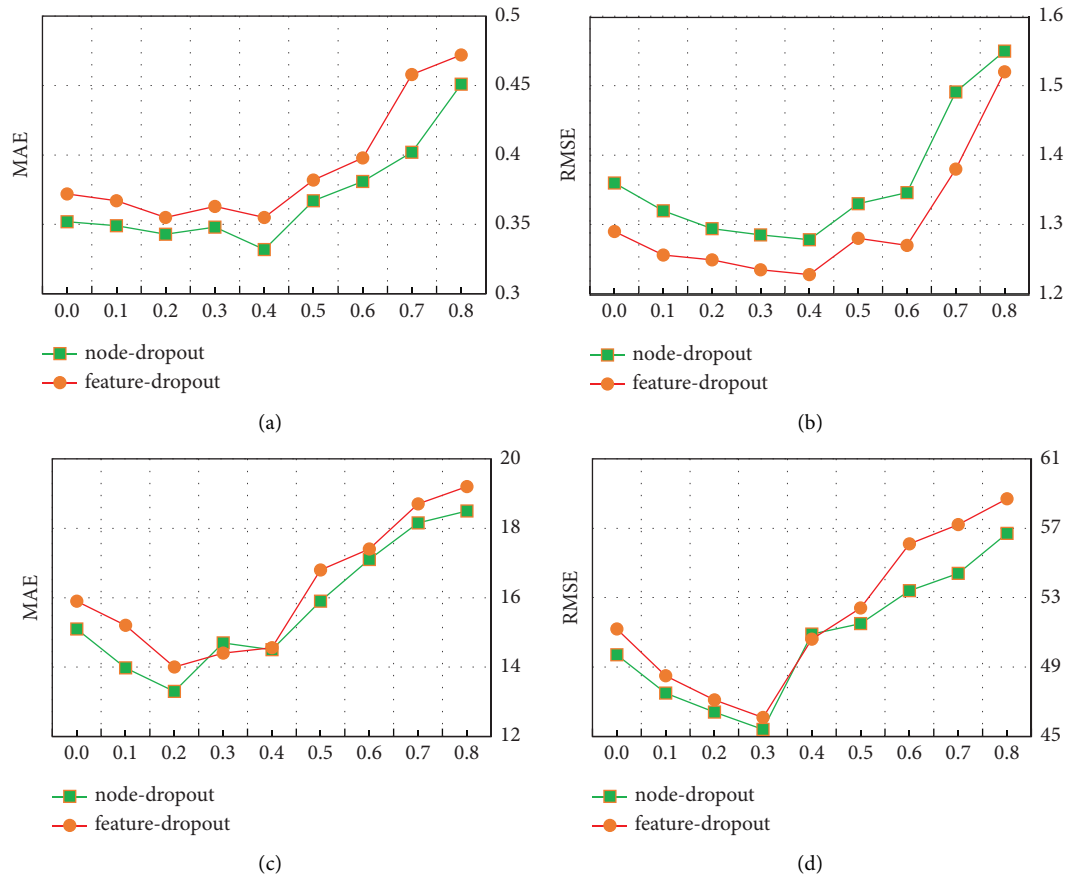


Figure 9: Comparison of dropout ratio. (a) Dropout ratio on RT. (b) Dropout ratio on RT. (c) Dropout ratio on TP. (d) Dropout ratio on TP.

Table 4: Runtime comparison of TP on average (hour).

Models	RT		TP	
	FSCM (h)	Non-FSCM (h)	FSCM (h)	Non-FSCM (h)
LDA	7.5	10.7	7.6	10.9
GraphSage	8.9	14.3	9.1	14.6
GCN	12.1	16.5	12.5	15.9
NGCF	14.7	18.2	14.9	18.4
RDHE	15.2	18.4	15.4	18.6
Average	**11.7**	**15.7**	**11.9**	**15.8**

clear that at the scale of our dataset, the fast similarity calculation method we designed is effective.

6. Conclusions and Future Work

This article proposes a novel community discovery and hierarchical embedding method for web service prediction based on a dual representation and similarity calculation mechanism, which can both utilizes attribute information and structural information to achieve a better prediction effect. We also propose a fast similarity calculation method to improve the algorithm's ability to handle large-scale data, which can avoid the curse of dimensionality of traversing all data samples and reduce the time complexity of similarity calculation. A huge amount of experiments had been carried out on the WS-Dream. The experimental results show that compared to the classic methods (NGCF, PinSage, KGAT, GraphRec, and NIA-GCN), the precision of our RDHE method is up to 11.11%, 9.19%, 9.19%, 5.62%, and 3.45% higher.

Although the experiments have verified the effectiveness of the community discovery method introduced in this article, our method is based on the multi-layer representation of Wide &Deep and GraphSage, the architecture is more complex, and it is based on the convenience of label acquisition, but it is actually a difficult task to obtain labels. According to our analysis and cognition, the next improvement directions are: (1) further research on semi-supervised and even unsupervised methods and (2) further research on end-to-end training, without sub-module or staged training in the learning process to directly optimize the tasks.

Data Availability

The WS-Dream dataset can be obtained at https://github.com/wsdream/wsdream-dataset.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work was supported by Key Projects of Natural Sciences Research in Anhui Universities of China (KJ2016A502), Key research projects of Chaohu University (XLZ-202106), the Key Research and Development Plan of Anhui Province, China (201904a05020091), Anhui Province Teaching Demonstration Course Project (2020SJJXSFK17), and Anhui Province Teaching Research Project (2019jyxm1187).

References

[1] S. A. McIlraith, T. C. Son, and H. Zeng, "Semantic web services," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 46–53, 2001.

[2] Y. Syu and C. M. Wang, "QoS time series modelling and forecasting for web services: a comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 18, 2021.

[3] S. Mokarizadeh, P. Küngas, and M. Matskin, "Utilizing web services networks for web service innovation," in *Proceedings of the 2014 IEEE International Conference on Web Services*, pp. 646–653, Anchorage, Alaska, July, 2014.

[4] Y. Cao, J. Liu, M. Shi, B. Cao, X. Zhang, and Y. Wang, "Relationship network augmented web services clustering," in *Proceedings of the 2019 IEEE International Conference on Web Services (ICWS)*, pp. 247–254, Milan, Italy, July, 2019.

[5] J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: social influence prediction with deep learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2110–2119, London, UK, August, 2018.

[6] M. Tang, Z. Zheng, G. Kang, J. Liu, Y. Yang, and T. Zhang, "Collaborative web service quality prediction via exploiting matrix factorization and network map," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 126–137, 2016.

[7] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 289–299, 2013.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, https://arxiv.org/abs/1609.02907.

[9] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, April, 2018.

[10] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[11] M. Wang, X. Zheng, Y. Yang, and K. Zhang, "Collaborative filtering with social exposure: a modular approach to social recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[12] W. Fan, Y. Ma, Q. Li et al., "Graph neural networks for social recommendation," in *Proceedings of the World Wide Web Conference*, pp. 417–426, Lyon, France, May, 2019.

[13] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 165–174, Paris, July, 2019.

[14] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, London, UK, July, 2018.

[15] Y. Xiao, J. Liu, G. Kang, and B. Cao, "LDNM: a general web service classification framework via deep fusion of structured and unstructured features," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3858–3872, 2021.

[16] R. A. H. M. Rupasingha, I. Paik, and B. T. G. S. Kumara, "Specificity-aware ontology generation for improving web service clustering," *IEICE - Transactions on Info and Systems*, vol. E101.D, no. 8, pp. 2035–2043, 2018.

[17] B. T. Kumara, I. Paik, T. H. A. S. Siriweera, and K. R. Koswatte, "QoS aware service clustering to bootstrap the web service selection," in *Proceedings of the 2017 IEEE International Conference on Services Computing (SCC)*, pp. 233–240, IEEE, Honolulu, HI, USA, June, 2017.

[18] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, Shenzhen, China, July, 2005.

[19] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[20] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *Proceedings of the ICLR*, Austria, April, 2015.

[21] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proceedings of the ICLR*, Vancouver, BC, Canada, August, 2014.

[22] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric Matrix Completion with Recurrent Multi-Graph Neural Networks," in *Proceedings of the NIPS*, Barcelona, Spain, December, 2017.

[23] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[24] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, San Francisco, CA, USA, August, 2016.

[25] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, New York USA, August, 2014.

[26] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation Learning on Graphs: Methods and Applications," *IEEE Data Engineering Bulletin*, vol. 40, no. 3, pp. 52–74, 2017.

[27] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, pp. 18–42, 2017.

[28] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 793–803, Anchorage AK USA, July, 2019.

[29] Y. Jacob, L. Denoyer, and P. Gallinari, "Learning latent representations of nodes for classifying in heterogeneous social networks," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 373–382, Tempe, AZ, USA, February, 2014.

[30] P. Pham and P. Do, "W-MMP2Vec: topic-driven network embedding model for link prediction in content-based heterogeneous information network," *Intelligent Data Analysis*, vol. 25, no. 3, pp. 711–738, 2021.

[31] Q. Yu, H. Wang, and L. Chen, "Learning sparse functional factors for large-scale service clustering," in *Proceedings of the 2015 IEEE International Conference on Web Services*, pp. 201–208, IEEE, New York USA, July, 2015.

[32] K. Zeng and I. Paik, "Semantic service clustering with lightweight bert-based service embedding using invocation sequences," *IEEE Access*, vol. 9, Article ID 54298, 2021.

[33] J. Xu, Z. Zhu, J. Zhao, X. Liu, M. Shan, and J. Guo, "Gemini: a novel and universal heterogeneous graph information fusing framework for online recommendations," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3356–3365, CA USA, August, 2020.

[34] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, "Kgat: knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 950–958, Anchorage, AK, USA, August, 2019.

[35] J. Sun, Y. Zhang, W. Guo et al., "Neighbour interaction aware graph convolution networks for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1289–1298, Xi'an, China, July, 2020.

[36] J. Yang and J. Leskovec, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 587–596, Tempe, AZ, USA, July, 2013.

[37] L. Lin, G. Wang, W. Zuo, X. Feng, and L. Zhang, "Cross-domain visual matching via generalized similarity measure and feature learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1089–1102, 2017.

[38] S. Eghbali and L. Tahvildari, "Fast cosine similarity search in binary space with angular multi-index hashing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 329–342, 2019.

[39] H. T. Cheng, L. Koc, J. Harmsen et al., "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 7–10, Boston, MA, USA, September, 2016.

[40] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[41] W. L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. J. Hsieh, "Cluster-gcn: an efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge*

Discovery & Data Mining, pp. 257–266, Anchorage AK USA, August, 2019.

[42] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[44] Z. Zheng, Y. Zhang, and M. R. Lyu, "Investigating QoS of real-world web services," *IEEE transactions on services computing*, vol. 7, no. 1, pp. 32–39, 2014.

[45] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4027667, 12 pages
https://doi.org/10.1155/2022/4027667

Research Article

BDS-3 Broadcast Ephemeris Orbit Correction Model Based on Improved PSO Combined with BP Neural Network

Jiebo Peng, Feng Liu, and Wenjin Hu

School of Computer Science, Xi'an Polytechnic University, Xi'an, Shaanxi, China

Correspondence should be addressed to Feng Liu; liufeng@xpu.edu.cn

Received 31 March 2022; Revised 24 August 2022; Accepted 28 August 2022; Published 26 September 2022

Academic Editor: Ripon Chakrabortty

During the operation of navigation satellites, errors in the broadcast ephemeris orbits are caused by the influence of ingress factors and other factors. To address this phenomenon, this paper examines the use of the computational intelligence (CI) methods to implement track correction and to develop an optimized BP neural network model based on an improved particle swarm algorithm. The model improves the inertia weights and learning factor parameters of the particle swarm optimization (PSO) algorithm to improve the global optimization capability and accelerate the convergence speed. The improved PSO (IPSO) algorithm is used to perform a global optimization search for the hyperparameters of the BP neural network, and then the neural network model is trained by broadcast ephemeris Keplerian root number and regression parameters. The model is validated using the broadcast ephemeris data of the BDS-3 MEO and IGSO satellites, and the mean square error correction rate of multiple satellites with different correction models shows that the error correction rate of the IPSO-BPNN model can reach 70.2–84% and the error correction rate can be improved by 14.2–56.8% compared with the PSO-BPNN model. The proposed algorithm provides an important reference for BDS-3 and other global navigation satellite systems for improving the accuracy of satellite orbit determination.

1. Introduction

The BeiDou satellite navigation system is a global satellite navigation system built and operated independently by China. In 2020, the third phase of the BeiDou system (BDS-3) was completed and it is now capable of providing high precision positioning and timing services to the world. The accuracy of the satellite ephemeris, which includes the broadcast ephemeris and postevent precision ephemeris, is directly tied to the accuracy of satellite positioning. The broadcast ephemeris has the feature of real time, but the accuracy is not sufficiently high. The precision ephemeris can be approximated as the true value of satellite coordinates with high accuracy, but the data have a high delay and cannot provide navigation and positioning services. Ensuring the accuracy of broadcast ephemeris has been a challenge that has been studied by many researchers [1–3]. The parameters of the broadcast ephemeris forecast can be optimized by combining mathematical and dynamical models to improve the positioning and timing accuracy. The uncertainty of the broadcast ephemeris spatial regimes and the high rate of change of the parameters make it difficult to improve the broadcast ephemeris accuracy by optimizing the dynamical model. In the process of analyzing the broadcast ephemeris orbit error data, it has been found that a new breakthrough approach can be identified from the error data itself by correlating the broadcast ephemeris parameters with the error data to reduce the broadcast ephemeris error and improve the accuracy of the broadcast ephemeris orbit [4, 5].

Broadcast ephemeris errors are time series type data, and numerous scholars have contributed to the prediction of time series data with good prediction results, which include the use of deep learning methods and neural network models. Rui Yan et al. compared the prediction performance in air quality prediction using BPNN, CNN, LSTM, and CNN-LSTM models and in the overall prediction, BPNN, CNN, LSTM, and CNN-LSTM models all have good results

in the overall prediction [6]. Wu et al. proposed a wave energy prediction model using an improved BPNN model and an improved ensemble empirical mode decomposition. The wave energy is predicted by this model, and finally, the validity of the prediction model is verified by using the actual measured wind waves of the real ocean as an example [7]. Wu et al. decomposed the original wind speed sequence into a set of intrinsic mode functions using variational mode decomposition and then combined the high-performance multiview prediction and interpretable temporal dynamic insight based on the attentional deep learning model. The method is eventually used to predict wind speed from meteorological data with perfect results [8]. Lv and Wang similarly proposed an improved hybrid time series decomposition strategy for effective wind speed prediction, and the test results showed better results compared with existing decomposition strategies, optimization techniques, and deep learning predictors [9]. The use of neural network models in satellite navigation data prediction has also been investigated by several scholars. Yaqi et al. [10] predicted the broadcast ephemeris orbit error with the particle swarm optimization (PSO)-BPNN model and applied the final prediction results to error compensation. The model shows a strong fitting ability and prediction effect on the broadcast ephemeris orbit error, according to the trials. Changdong et al. [11] corrected the BDS satellite clock aberration using a neural network and improved the prediction accuracy compared with that of the traditional correction model. Bohua et al. [12] used a convolutional neural network to identify the outliers of BDS satellite clock aberration data to greatly improve the accuracy of removing the outliers of the clock aberration data. Ye et al. [13] suggested a method for predicting satellite clock aberration in the medium and long run. However, this method does not effectively address the prediction accuracy of the satellite in the short term.

Neural network models are simulated neuronal structures that are self-learning and self-adaptive and have significant effects for some nonlinear mapping relations [14, 15]. The backpropagation (BP) neural network is one of the feed-forward multilayer perceptrons, whose training process is a model that reverses the expected output and the actual output as error for data prediction by the backpropagation neural networks (BPNNs) [16]. The BP neural network is a neural network model that propagates backward according to the error value and has the advantages of a simple model, fast training, and self-adaptation compared with CNN and RNN when dealing with one-dimensional data of broadcast ephemeris error. Although the BP neural networks have a strong learning ability, the setting of initial weights affects the convergence speed and performance of the model [17, 18]. The classic particle swarm algorithm (PSO) is an intelligent optimization algorithm that simulates a flock of foraging birds by iterating the position and velocity of the particles in the swarm through the fitness function to solve for the optimal particles and eventually apply the optimized initial weight values to the BP neural network. The advantages of the PSO algorithm are its strong global search capability, simple structure, and easy implementation. However, it also has some disadvantages; for example, the setting of hyperparameters can affect the convergence results of the algorithm and it is easy to fall into local optimal solutions [19]. One of the drawbacks of the particle swarm algorithm will be addressed by intervening in the selection of hyperparameter values to make its algorithm converge faster while avoiding the emergence of local optimal solutions. This paper will make three main contributions:

(a) Improvement of inertia weights in the PSO algorithm. By adopting the dynamic adjustment of the inertia weights with a nonlinear function, the global optimization-seeking ability of the PSO algorithm is enhanced to avoid the emergence of local optimal solutions.

(b) Improvement of learning factors in the PSO algorithm. With the depth of iteration, different learning factors are selected to make the model converge faster.

(c) Apply the IPSO-BP model to the BDS-3 satellite broadcast ephemeris error correction. The validity of the model is verified by the latest BDS satellite broadcast ephemeris data.

2. Broadcast Ephemeris Data Error Analysis

2.1. Error Data Preprocessing. BeiDou satellite data are used as experimental data in this work. China constructed and operates the BeiDou satellite navigation system, which is a global satellite navigation system. The BeiDou system's third phase (BDS-3) was finished in 2020 and it is now capable of providing high-precision location and timing services to the world. The satellite ephemeris generally includes the broadcast ephemeris, ultrafast orbit, fast orbit, and final orbit ephemeris data. The ultrafast orbit data have a delay of 3 h and a sampling time of 15 min. The fast orbit data are delayed by 17 h and sampled for 15 min. The final orbit data are delayed for 12 d and sampled for 15 min. The final orbits can be approximated as real orbit data [20].

The BDS-3 broadcast ephemeris contains six Keplerian orbital roots and nine ingress parameters for all satellites observed by the current observatory under the toe reference ephemeris, and the satellite position and velocity at time "t" under the "toe" reference ephemeris can be calculated based on the formula of the broadcast ephemeris for calculating the satellite position [21].

The receiver independent exchange (RINEX) format satellite ephemeris data in this research are provided by the iGMAS Wuhan station. As real-time satellite position data, the BDS broadcast ephemeris is sampled once every hour. Under the reference of the same satellite in the same ephemeris, the BDS broadcast ephemeris orbit error can be obtained by determining the difference between the real-time position calculated by the broadcast ephemeris and the final orbit ephemeris. The calculation formula is as follows:

$$\begin{cases} \Delta x = x_f - x_p, \\ \Delta y = y_f - y_p, \\ \Delta z = z_f - z_p, \end{cases} \tag{1}$$

where Δx, Δy, and Δz represent the errors in the x-, y-, and z-directions, respectively; x_f, y_f, and Z_f represent the broadcast ephemeris under the x, y, and z coordinate values, respectively; and x_p, y_P, and z_p represent the values of the x, y, and z coordinates, respectively, in the final orbital ephemeris.

2.2. Error Data Analysis. Without loss of generality, this work estimates the broad-cast ephemeris orbit error data for the BDS-3's IGSO and MEO satellites over a continuous period from April 20, 2022, to April 30, 2022. Among these satellites, C19, C20, and C21 are the BDS-3 MEO satellites, and C38, C39, and C40 are the BDS-3 IGSO satellites. Figure 1 shows the orbital errors of the three MEO satellites in the x-, y-, and z-directions, which are all in the range of ±1.5 m. Figure 2 shows that the orbital errors of the three IGSO satellites in the y-direction are between 0 and 2 m, while the errors in the x- and z-directions are in the range of ±2 m and noisy data are present. According to the trend of the error curve in the figure, it is observed that there are different oscillation patterns in each direction, but these patterns cannot be completely eliminated by the current dynamical model.

During the actual operation of satellites, they are affected by many different factors related to their natural environment, including the nonspherical gravity of the Earth, N-body uptake, solid tides, and ocean tides. [22]. The presence of these factors causes the broadcast ephemeris orbit to deviate from the actual orbit, thus producing errors. Theoretically, the accuracy of the models of these regression components should have a direct impact on the accuracy of the broadcast ephemeris orbits; however, the development of correct regression models has recently faced some difficulties.

In addition to the six Keplerian roots ($\sqrt{a}$, $[ee]$, Ω_0, ω, i_0, and M_0), the BDS-3 broadcast ephemeris orbit data contain nine ingress parameters ($\dot{I}$, $\dot{\Omega}$, Δn, C_{us}, C_{uc}, C_{is}, C_{ic}, C_{rs}, and C_{rc}) [21], as shown in Table 1. The satellite positions and velocities are calculated from these uptake parameters. Therefore, the ingress parameters are crucial for errors in the broadcast ephemeris orbit, and they also act as influencing factors that cause the broadcast ephemeris orbit to deviate from the actual orbit. These factors are fully utilized to study the relationship between them and the broadcast ephemeris orbit error and then the nonlinear mapping relationship between them is found with the BPNN model.

3. Introduction to the Broadcast Ephemeris Orbit Correction Model

3.1. BPNN-Based Broadcast Ephemeris Orbit Correction Model. The BPNN model is a backward-updating neural network model that is based on the difference between the actual and expected output signal values. This expected signal value is based on the principle of most rapid gradient descent. Theoretically, a three-layer BPNN can complete the mapping relationship in any dimension. In this paper, a three-layer BPNN model is used, and its structure is shown in Figure 3. The BPNN is continuously updated with the number of rounds of network training iterations, the output error of the BPNN slowly decreases and the optimal solution is finally obtained [23]. The process of reverse updating the BPNN is shown in Figure 4.

The input parameters in the BP three-layer network model are the reference moment, Keplerian orbit root number, and regenerative parameters, for a total of 16 parameters. The output represents the satellite's error in the x-, y-, and z-directions.

3.2. IPSO-BPNN Real-Time Correction Model

3.2.1. Particle Swarm Optimization (PSO). PSO is a computational intelligence method that simulates a flock of bird foraging activity by calculating the distance and speed between individual particles to optimize the population as a whole. [24]. The core algorithmic idea is that in D-dimensional space, there are N particles, and each of the N particles is searching for its individual position and group position to reach the optimum. The running speed of the particles is $V^t_{[ii],j}$, and each particle finds its optimum position within the group position after many iterations of each individual position and group position to constrain the movement speed. The traditional PSO algorithm is formulated as follows:

$$x^t_{i,j} = \omega v^t_{i,j} + c_1 r_1\left(p^t_{i,j} - x^t_{i,j}\right) + c_2 r_2\left(p^t_{g,j} - x^t_{i,j}\right), \tag{2}$$

$$x^{t+1}_{i,j} = x^t_{i,j} + v^{t+1}_{i,j}, \tag{3}$$

where ω represents the inertia weight; c_1 and c_2 represent learning factors; r_1 and r_2 represent random numbers 0 and 1; and $V^t_{i,j}$, $x^t_{[ii],j}$, $p^t_{g,j}$ and $p^t_{[ii],j}$ represent the speed, location, group optimum, and individual optimum, respectively.

The traditional particle swarm algorithm faces a problem in that the inertia weights in equation (2), c_1 and c_2 are learning factors that can lead to instability and early convergence problems in the algorithm [25]. Therefore, in recent years, some scholars have proposed many particle swarm algorithms, such as the SPSO [26] and QPSO [27] algorithms. Among them, the SPSO algorithm changes the search space from a real-valued space to a set of selected-valued spaces, which is optimized by reducing the computational complexity of the fitness function. The QPSO algorithm optimizes the PSO algorithm by updating the particle positions according to the average optimal position of the particles. The IPSO algorithm in this paper will intervene by selecting values of hyperparameters and using nonlinear functions to dynamically adjust the values of inertia weights and learning factors to make its algorithm converge faster while avoiding the emergence of local optimal solutions. In addition, the IPSO algorithm also has the advantage of simple results and easy implementation. Therefore, it is more suitable for the optimization of the BPNN network models.

3.2.2. Improvement in the Inertia Weights in PSO. At various stages of convergence, the IPSO algorithm improves the determination of inertia weight values and learning factor

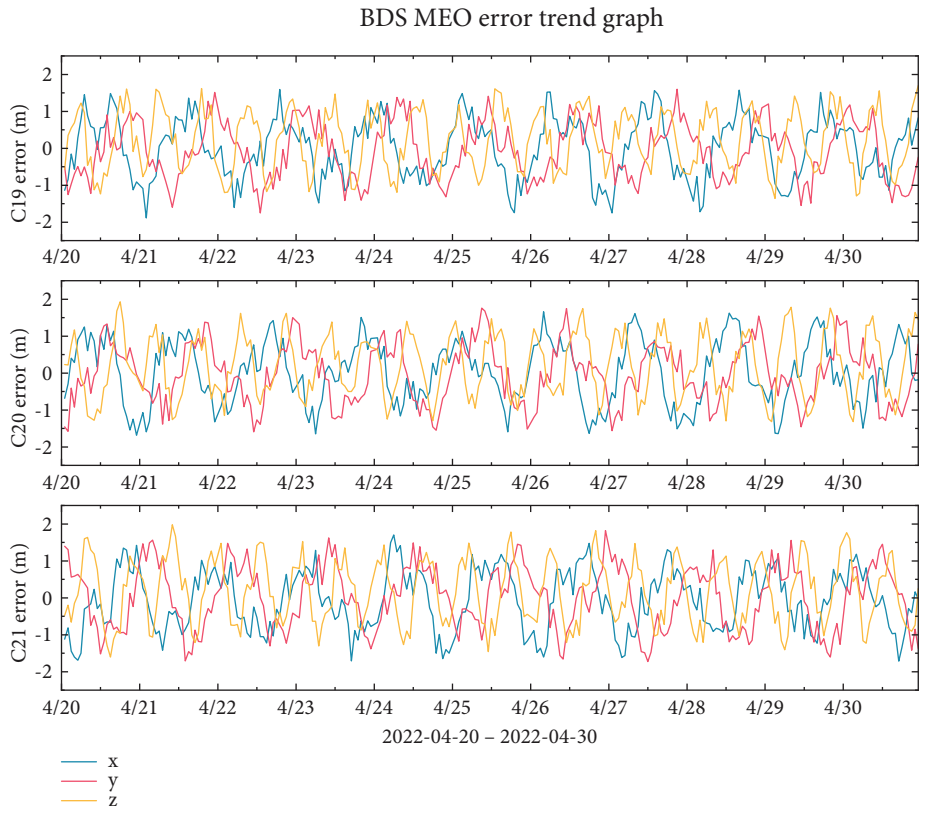


Figure 1: BDS-3 MEO broadcast ephemeris error trend chart.

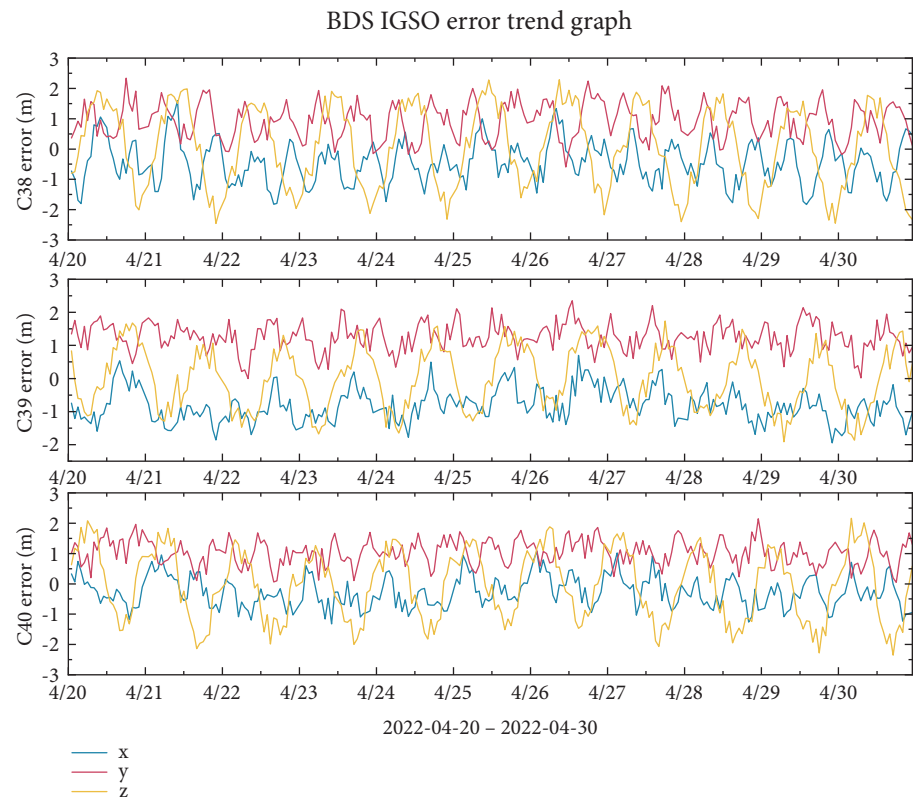


Figure 2: BDS-3 IGSO broadcast ephemeris error trend chart.

Table 1: Broadcast ephemeris camera parameters description table.

Parameters	Description	Unit
$\dot{I}$	Rate of change in orbital inclination	rad/s
$\dot{\Omega}$	Rate of change in the equatorial diameter of the ascending intersection	rad/s
Δn	Correction value of the angular velocity at the level of proximity	rad/s
C_{us}, C_{uc}	Amplitudes corrected by the sine and cosine summation terms of the latitude amplitude angles	rad
C_{is}, C_{ic}	Amplitude of the correction of the sine and cosine summation terms of the orbital inclination angle	rad
C_{rs},C_{rc}	Amplitude of correction of the sine and cosine summation terms of the orbital radius	m

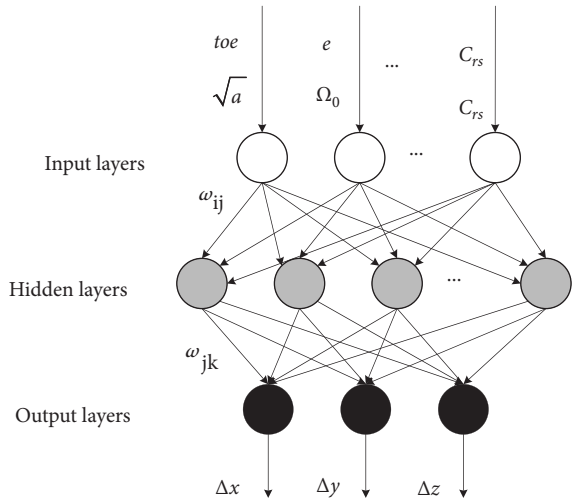


Figure 3: BPNN model diagram.

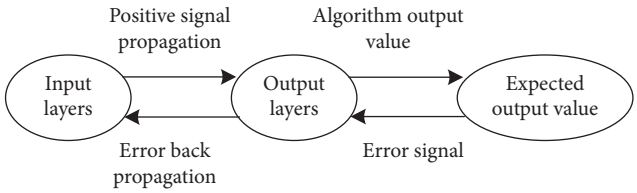


Figure 4: BPNN learning process.

values. [28]. According to the experimental investigation, the value of ω in this algorithm has a significant impact on the overall performance of the population search; a lower ω value corresponds to greater ease of obtaining a local optimum, while a greater ω value makes it easier to achieve the global optimum. As a result, a larger ω value should be chosen for global optimization at the beginning of the algorithm iteration, and a lower ω value should be chosen for local optimization at the end of the algorithm iteration. Since the curve of the sinusoidal function in the $[0, \pi]$ interval first rises and then falls, combined with the selection of the ω parameter to first use a smaller value for the motion, it gradually expands the range explored by PSO and enhances the global search capability, then continuously updates the global optimal value, and finally, reduces the inertia weight to enhance the local search capability, accelerating the convergence of PSO. The inertia weight improvement formula is as follows:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \cdot \sin\left(\frac{\pi t}{T}\right), \tag{4}$$

where $\omega_{\max}$, $\omega_{\min}$, and T are the maximum weight, minimum weight, and maximum number of iterations, respectively.

For $\omega_{\min}$ of 0.4 and $\omega_{\max}$ of 0.8, the curve for ω plotted versus time is shown in Figure 5.

3.2.3. Improvement in the Learning Factor in PSO. To address the drawback of the premature convergence of PSO, the calculation of the learning factor is improved; here, a larger (smaller) value of c_1 makes it easier (harder) for the individual position of the particles to achieve superiority, while a larger (smaller) value of c_2 makes it easier (harder) for the group position of the particles to achieve superiority. To enable the PSO algorithm to rapidly identify the global extremal solution, c_1 and c_2 can be adjusted as follows:

$$\begin{cases} c_1 = \dfrac{(c_s - c_e) \bullet (T - t)}{T} + c_e, \\ c_2 = 4 - c_1, \end{cases} \tag{5}$$

where the values of c_s and c_e are between 1 and 4.

For c_s taken as 2.5 and c_e taken as 0.5, the obtained c_1 and c_2 are plotted versus time in Figure 6.

3.2.4. Parameter Determination of the IPSO-BPNN Model. The beneficial effect of the BPNN model depends strongly on the setting of the network model parameters and is prone to suffer from drawbacks such as trapping in local optima and slow convergence. Improved particle swarm optimization, on the other hand, can optimize population data and discover the best solution in the population particles. In this paper, IPSO is used to choose all hyperparameters of the BPNN model to optimize the BPNN, prevent the model from reaching local optima, improve the BPNN's learning ability, and accelerate convergence.

(1) BPNN Parameter Determination. To ensure that the model can accurately map the relationship between the forecast orbital parameters and the errors of the BDS-3 broadcast ephemeris, the reference moment, the Keplerian orbital root number and regenerative parameters in the forecast orbital data are selected as the inputs of the IPSO-BPNN model, for a total of 16 parameters. The model outputs are Δx, Δy and Δz. m are set to 16, n is set to 3, and l is the hidden layer.

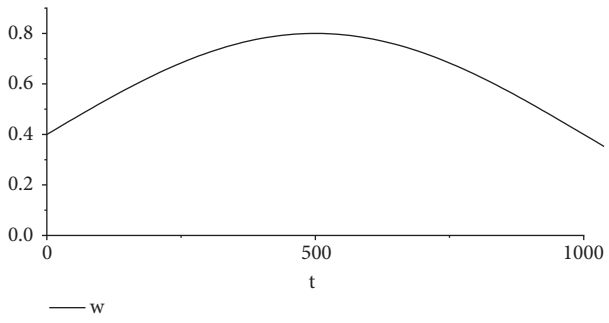


Figure 5: ω as a function of time.

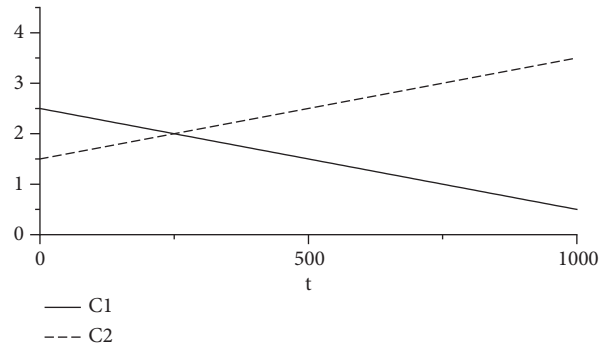


Figure 6: Learning factor evolution with time.

$$l = \sqrt{m+n} + a, a \in [1, 10]. \tag{6}$$

The training objective is 0.001 and the learning rate is 0.01. The number of model iterations is 1000, the training target is 0.001, and the learning rate is 0.01.

(2) IPSO Parameters and Fitness Function Settings. The purpose of IPSO is to find each hyperparameter suitable for the BPNN so that the population dimension D is the number of all hyperparameters in the neural network; then, the algorithm randomly generates the set of all particles. $p_{g,j}^t$ and $p_{i,j}^t$ are initialized in PSO and c_s and c_e are initialized for the learning factor.

The number of example populations is 250; the values of the learning factors c_s and c_e affect the values of c_1 and c_2. Here, c_s and c_e are taken as 2.5 and 0.5, respectively, and the inertia weights $\omega_{\max}$ and $\omega_{\min}$ are taken as 0.8 and 0.4, respectively.

The fitness function can help the particle adjust the particle position and velocity, and its *mapping* relationship is an inverse proportional function, where the fitness function is given as the following equation:

$$f = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[f(x_i) - y_j\right]^2}, \tag{7}$$

where n, $f(x_i)$, and y_i represent the number of samples trained, the PSO predicted output value and the PSO expected output value, respectively.

3.2.5. IPSO-BPNN Algorithm Flow. The BDS-3 broadcast ephemeris data in a continuous time segment are prepared, and the satellite forecast position is calculated using the formula for the satellite position and velocity. Then, the orbit error data are calculated using the satellite precision orbit data. Since the parameters of the broadcast ephemeris data are not at the same order of magnitude, it is not possible to use the neural network for training, and the data must be normalized to eliminate the differences in the magnitude. The parameters are mapped to the range between 0 and 1 using the normalization formula.

$$y = \frac{(x - \overline{X})}{(x_{\max} - x_{\min})}, \tag{8}$$

where y, $\overline{X}$, $x_{\max}$, and $x_{\min}$ are the mapped value, data mean, maximum value in the sample, and minimum value in the sample, respectively.

Then, the output of the IPSO algorithm is utilized as the training input for the BPNN, the preprocessed data are fed into the network for model training, and the IPSO-BPNN model's error prediction value is rectified based on the original error. The detailed process of the IPSO-BPNN satellite orbit error-correction model is shown in Figure 7.

4. Results and Discussion

In this paper, the ephemeris data of all BDS-3 satellites are selected for a total of 22 days from April 20, 2022, to May 12, 2022. According to the broadcast ephemeris orbit data update frequency of once per hour, a total of 528 data items are calculated for a single satellite after the original data. The number of satellites used to collect the data and the number of data items are shown in Table 2. The data are divided into two parts according to time, where the data from April 20 to May 9, 2022 are used for model training and the data from May 10 to May 12 are used for model testing, and then the results of the different modified models are compared.

To ensure that the IPSO-BPNN correction model is feasible, the BPNN, PSO-BPNN, and IPSO-BPNN correction models are compared and evaluated. Meanwhile, to verify the feasibility of other models on BDS-3 satellite broadcast ephemeris data, the LSTM models are added for comparison experiments. There are four models and their experimental data are used for both BDS-3 MEO and IGSO satellite ephemeris data.

4.1. BDS-3 MEO Error Correction Results. The error profiles of the BDS-3 MEO satellites C19 and C20 are shown in Figures 8 and 9, with true errors of less than 2 m in all three orientations. The graph shows the change curves of the BP model, PSO-BP model, IPSO-BP model, and LSTM comparison model after the error correction within 3 days, and it can be seen from the model comparison curves that the repair effect of LSTM is more similar to the BP model but less stable than the BP model, and the repair ability of these two models is not very obvious. The results of the PSO-BP and IPSO-BP models are more similar, but the IPSO-BP model is more stable than the PSO-BP model.

Tables 3and 4 show the mean, mean squared error, and standard deviation of the BPNN, PSO-BPNN, and IPSO-BPNN models for MEO satellites in the x-, y-, and z-directions. The results indicate that the error repairability of

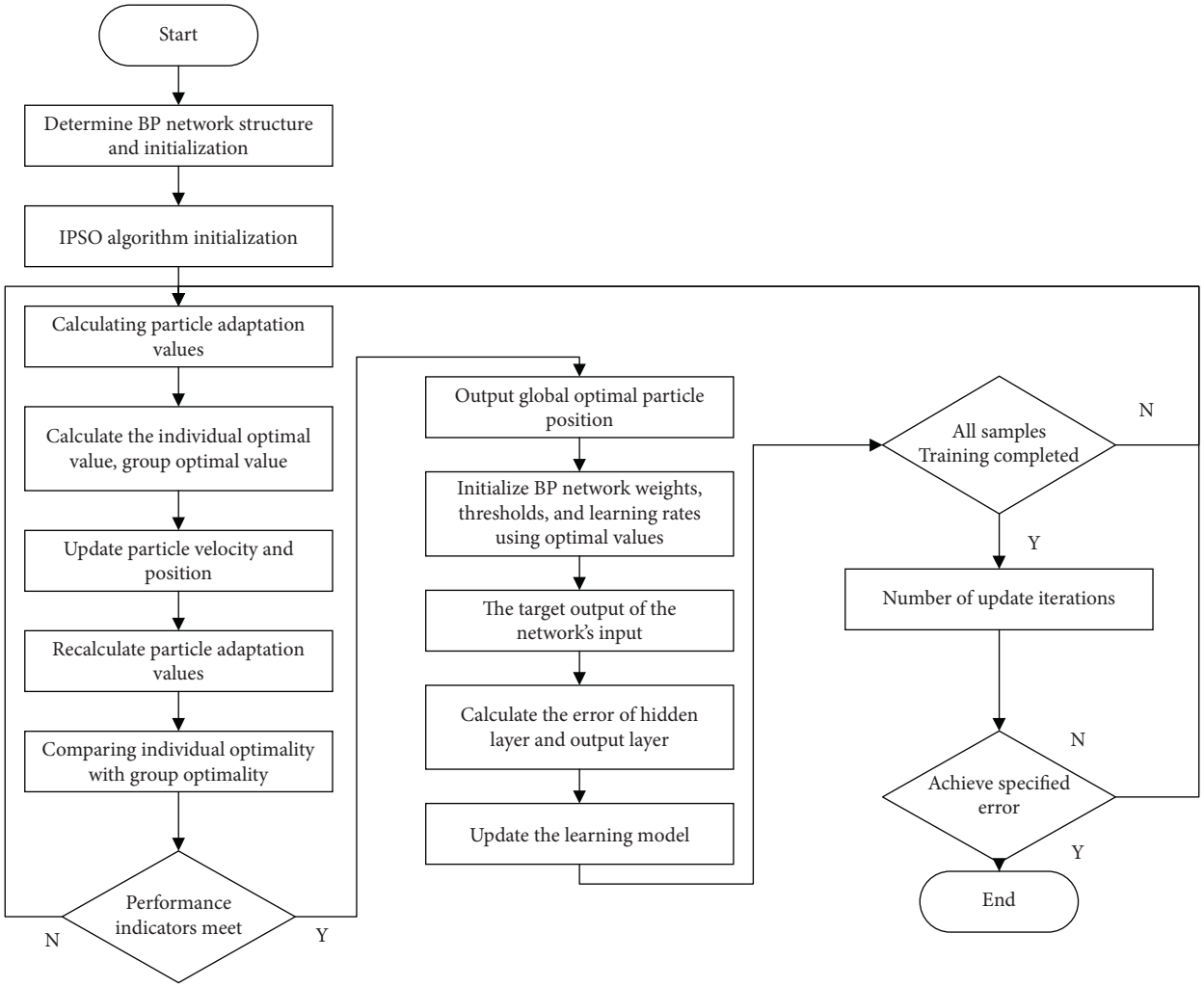


Figure 7: IPSO-BPNN flowchart.

Table 2: Datasets description table.

Parameters	Broadcast ephemeris data	Precision ephemeris data
PRN	BDS-3 (C19–C45)	BDS-3(C19–C45)
Date	2022-04-20 01 : 00~2022-05-12 23 : 00	2022-04-20 01 : 00~2022-05-12 23 : 00
File type	xxxx.rnx.gz	xxxx.SP3.gz
Lines per file	Approximately 135000	Approximately 11650
File size	Approximately 10000 KB	Approximately 700 KB
Number of files	22	22
Parsing format	RINEX format	SP3 format

the LSTM model is approximately 15%, while that of the BP model is approximately 34%. It is obvious that the BP model has better repairability than LSTM. However, in the comparison of the BP model, PSO-BP model and IPSO-BP model, the IPSO-BP model is better for broadcast ephemeris orbit error correction. The RMS error and SDT in the *x*, y, and *z* directions for the IPSO-BPNN model are decreased by 8–24 percent compared to the PSO-BPNN model.

4.2. BDS-3 IGSO Error Correction Results. Compared with those of the three generations of BDS-3 MEO satellites, the curves of the IGSO errors in the *x*-, *y*-, and *z*-directions are distinctly different, with no obvious regularity. As shown in Figures 10 and 11, the error range of C38 is between −2 m~1 m in the *x*-direction, −0.5 m~2 m in the *y*-direction, and −2.5 m~2.5 m in the *z*-direction. The trend of the curves in the figure shows that the LSTM model has the worst effect among all models in terms of repair ability, while the BP model has some correction effect, but the results are not ideal. The IPSO-BP model results are better than those of the PSO-BP model and its output is closer to 0 m. The trend of the error curve in star C39 changes again, but the IPSO-BPNN model's correction performance in the *x*-, *y*-, and *z*-directions is better than that of the PSO-BPNN model.

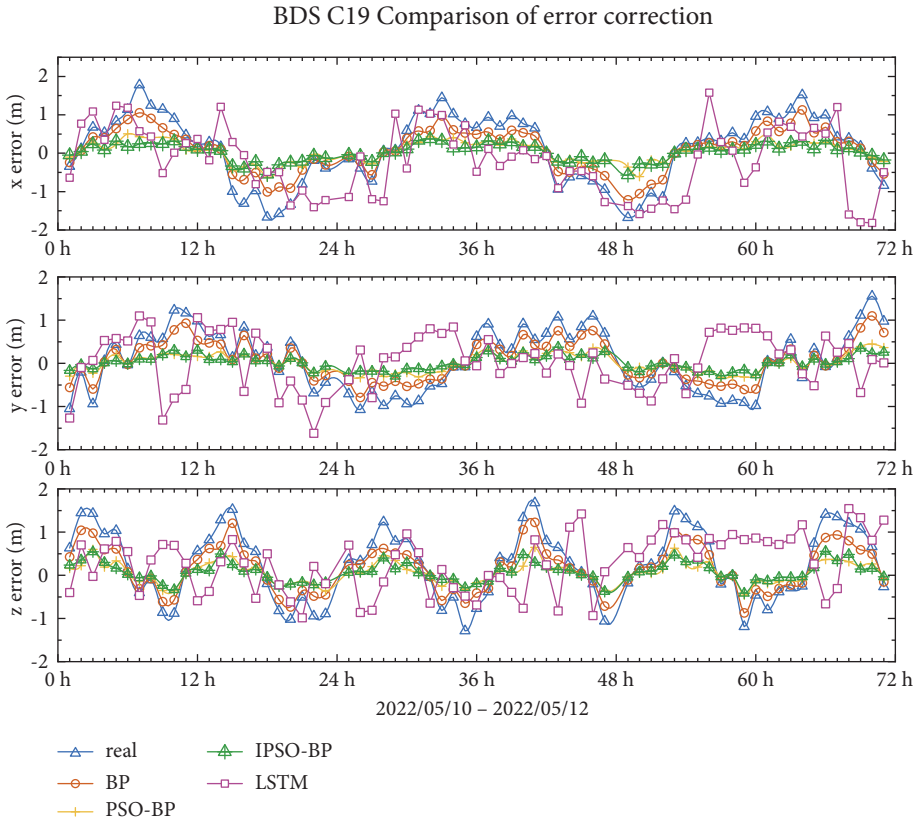


Figure 8: C19 broadcast ephemeris orbit error-correction curve.

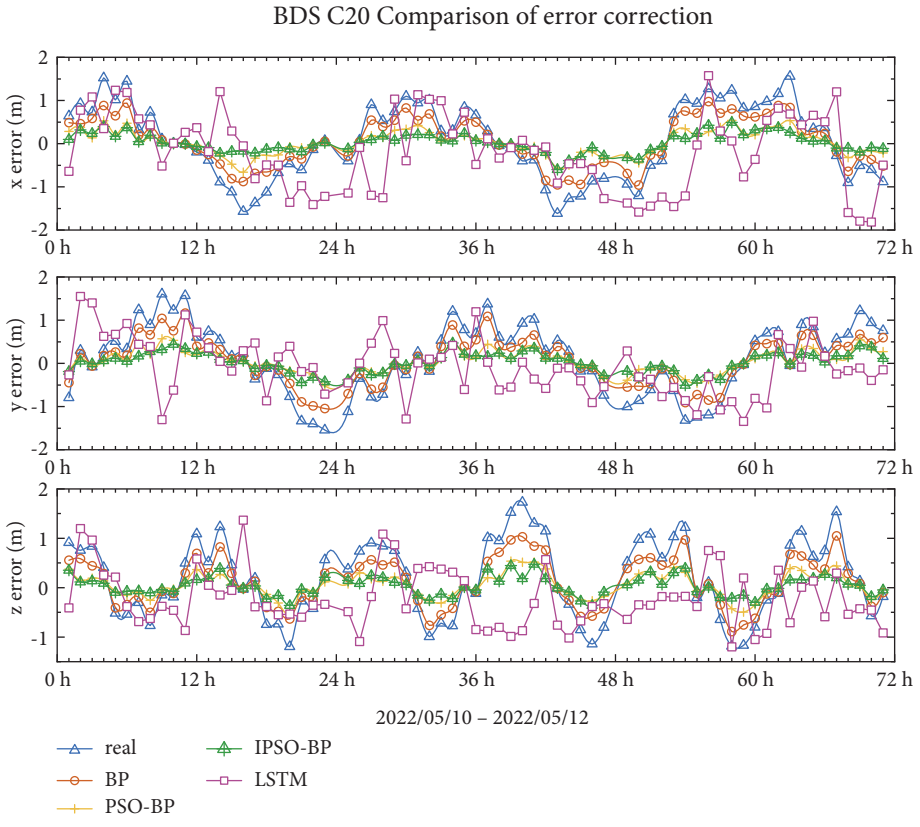


Figure 9: C20 broadcast ephemeris orbit error-correction curve.

Table 3: C19 broadcast ephemeris orbit error-correction table.

Model	x			y			z		
	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m
Real	0.729	0.875	0.869	0.597	0.698	0.693	0.681	0.802	0.796
BPNN	0.467	0.564	0.560	0.390	0.458	0.454	0.447	0.531	0.527
PSO-BPNN	0.220	0.270	0.268	0.176	0.209	0.208	0.198	0.246	0.244
IPSO-BPNN	0.184	0.225	0.223	0.137	0.167	0.166	0.176	0.223	0.222
LSTM	0.749	0.891	0.885	0.528	0.666	0.661	0.475	0.593	0.588

Table 4: C20 broadcast ephemeris orbit error-correction table.

Model	x			y			z		
	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m
Real	0.735	0.854	0.848	0.671	0.799	0.793	0.674	0.778	0.772
BPNN	0.486	0.564	0.560	0.445	0.536	0.532	0.429	0.499	0.496
PSO-BPNN	0.228	0.279	0.277	0.205	0.254	0.252	0.200	0.247	0.245
IPSO-BPNN	0.172	0.215	0.213	0.187	0.232	0.230	0.158	0.195	0.193
LSTM	0.649	0.782	0.777	0.537	0.650	0.645	0.571	0.669	0.664

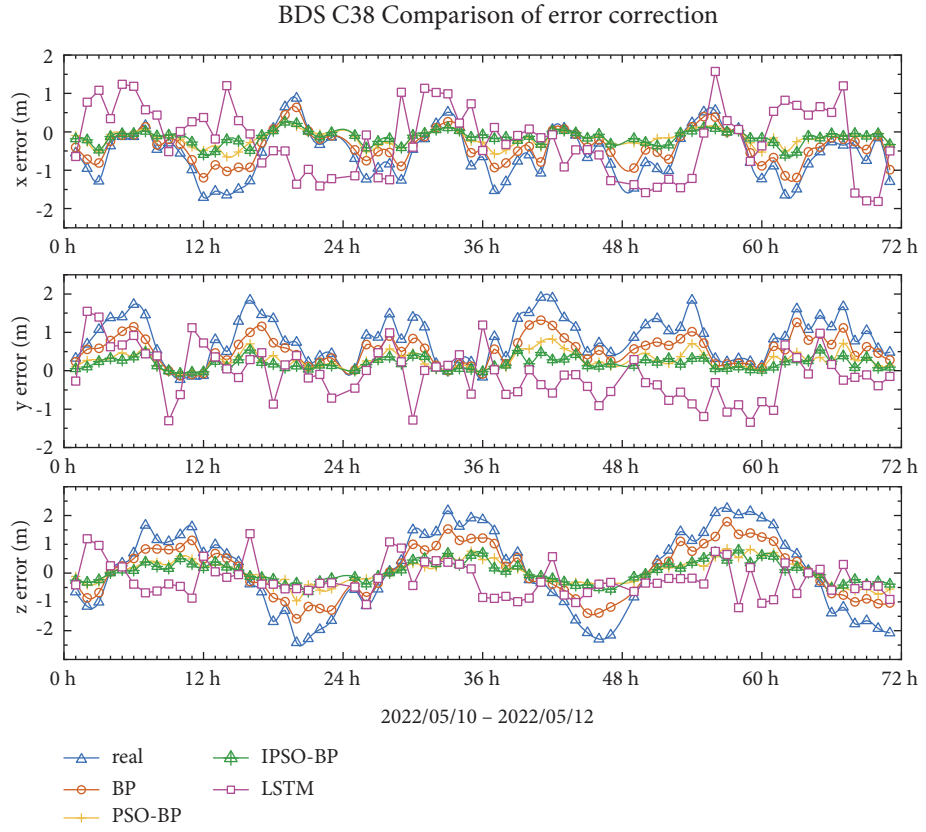


Figure 10: C38 broadcast ephemeris orbit error-correction curve.

Tables 5 and 6 demonstrate that among the three error correction models for the IGSO satellite, the IPSO-BPNN model is the most effective. This model's RMS error correction rate in the x-, y-, and z-directions is 62–74%. The IPSO-BPNN model's RMS and SDT in the x-, y-, and z-directions are lowered by 6% to 28%, respectively, compared to the PSO-BPNN model, considerably improving the accuracy of the broadcast ephemeris.

4.3. Error-Correction Results for All BDS-3 Satellites. To verify the feasibility of this model for other BDS-3 satellites, multiple broadcast ephemeris datasets are tested, and the corrected spatial distance values of broadcast ephemeris orbits are compared to the corresponding posterior precision ephemeris orbits. The correction rate of the mean square error of the orbit errors before and after the correction of multiple satellites is compared, and the model's

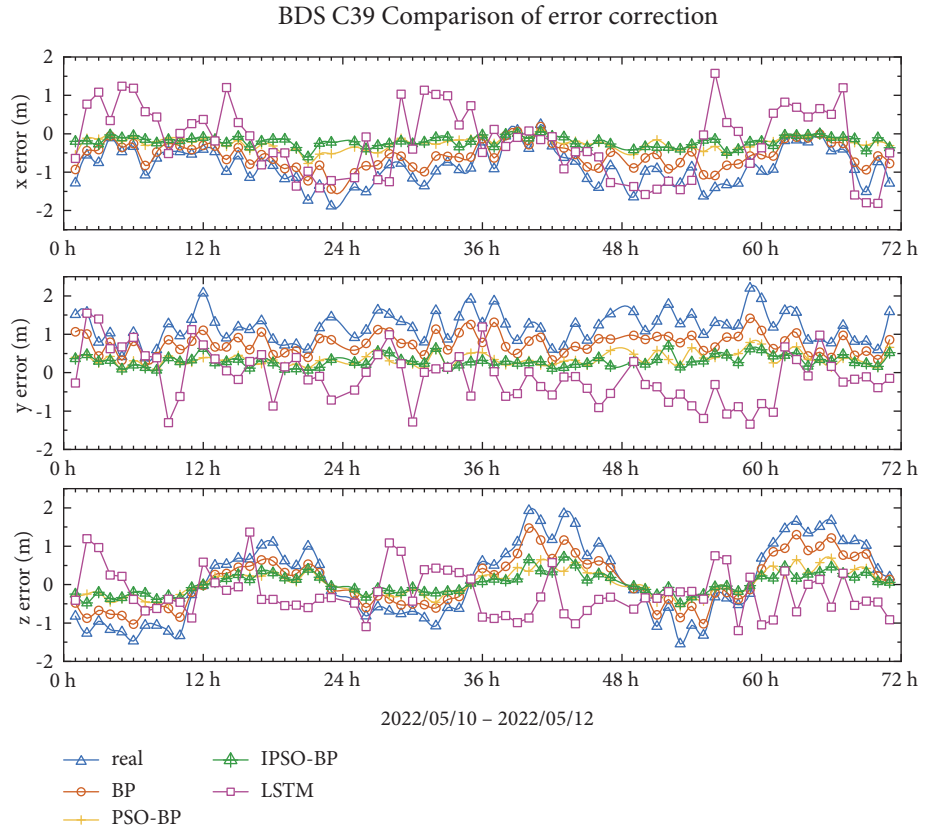


Figure 11: C39 broadcast ephemeris orbit error-correction curve.

Table 5: C38 broadcast ephemeris orbit error-correction table.

Model	x			y			z		
	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m
Real	0.519	0.627	0.623	0.480	0.572	0.567	1.206	1.390	1.380
BPNN	0.346	0.423	0.420	0.304	0.371	0.368	0.777	0.890	0.884
PSO-BPNN	0.164	0.202	0.200	0.168	0.210	0.209	0.347	0.414	0.411
IPSO-BPNN	0.139	0.182	0.180	0.126	0.151	0.150	0.304	0.361	0.358
LSTM	0.510	0.628	0.624	0.333	0.421	0.418	1.179	1.380	1.370

Table 6: C39 broadcast ephemeris orbit error-correction table.

Model	x			y			z		
	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m	Mean/m	STD/m	RMS/m
Real	0.385	0.473	0.470	0.316	0.396	0.393	0.881	0.993	0.986
BPNN	0.262	0.328	0.326	0.215	0.258	0.256	0.568	0.653	0.648
PSO-BPNN	0.131	0.165	0.164	0.125	0.160	0.159	0.283	0.329	0.327
IPSO-BPNN	0.109	0.135	0.134	0.116	0.149	0.148	0.226	0.271	0.269
LSTM	0.390	0.489	0.486	0.284	0.353	0.350	0.623	0.780	0.774

feasibility is validated by comparing the correction rate of the RMS of the orbit errors before and after the correction.

The test results are calculated in the three directions after the correction with equation (9). The calculation formula is as follows:

$$d = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}, \tag{9}$$

where Δx, Δy, and Δz represent the mean squared error in the x-, y-, and z-directions, respectively.

The results are shown in Table 7.

Based on the mean square error correction rates of multiple BDS-3 satellites for the different correction models, compared with the BP model, the LSTM model improves the error correction rate of the broadcast ephemeris orbit for

Table 7: Broadcast ephemeris orbit error-correction table.

Satellite type	PRN	RMS/m					Improvement rate			
		Real	LSTM	BPNN	PSO-BPNN	IPSO-BPNN	LSTM	BPNN	PSO-BPNN	IPSO-BPNN
MEO	C19	0.791	0.658	0.518	0.242	0.205	16.90%	34.6%	69.5%	74.1%
	C20	0.810	0.682	0.533	0.260	0.214	15.80%	34.2%	67.9%	73.6%
	C21	0.818	0.705	0.531	0.259	0.145	13.80%	35.1%	68.3%	82.3%
	C22	0.814	0.659	0.534	0.246	0.156	19.00%	34.3%	69.7%	80.8%
	C23	0.854	0.718	0.560	0.258	0.137	16.00%	34.5%	69.8%	84.0%
	C24	0.849	0.706	0.562	0.277	0.143	16.80%	33.8%	67.3%	83.2%
	C25	0.737	0.63	0.480	0.322	0.213	14.50%	34.9%	56.3%	71.1%
	C26	0.714	0.695	0.467	0.221	0.119	2.70%	34.6%	69.0%	83.4%
	C27	0.722	0.639	0.467	0.224	0.161	11.50%	35.3%	68.9%	77.7%
	C28	0.724	0.63	0.473	0.341	0.123	12.90%	34.7%	52.9%	83.1%
	C29	0.723	0.616	0.465	0.339	0.198	14.80%	35.7%	53.1%	72.6%
	C30	0.674	0.663	0.438	0.211	0.181	1.60%	35.0%	68.7%	73.1%
	C32	0.797	0.737	0.515	0.301	0.130	7.60%	35.3%	62.2%	83.7%
	C33	0.809	0.641	0.527	0.246	0.145	20.70%	34.8%	69.6%	82.1%
	C34	0.702	0.642	0.458	0.422	0.120	8.50%	34.8%	39.9%	82.9%
	C35	0.718	0.574	0.466	0.222	0.151	20.00%	35.0%	69.1%	79.0%
	C36	0.744	0.641	0.484	0.367	0.198	13.80%	34.9%	50.6%	73.4%
	C37	0.773	0.701	0.516	0.246	0.135	9.30%	33.2%	68.3%	82.5%
	C41	0.594	0.525	0.386	0.184	0.106	11.60%	35.0%	69.1%	82.1%
	C42	0.602	0.485	0.394	0.221	0.152	19.40%	34.5%	63.3%	74.7%
	C43	0.673	0.596	0.436	0.212	0.118	11.50%	35.2%	68.6%	82.4%
	C44	0.729	0.625	0.480	0.321	0.172	14.30%	34.1%	56.0%	76.4%
	C45	0.773	0.694	0.507	0.239	0.198	10.30%	34.4%	69.1%	74.4%
IGSO	C38	0.863	0.814	0.555	0.281	0.121	5.60%	35.7%	67.4%	86.0%
	C39	0.621	0.469	0.422	0.211	0.123	24.50%	32.0%	65.9%	80.2%
	C40	0.618	0.606	0.413	0.209	0.122	2.00%	33.1%	66.1%	80.2%

MEO satellites from 2.7% to 20.7% and the error correction rate of the broadcast ephemeris orbit for IGSO satellites from 2% to 24.5%. The model stability is poor.

In the BP model, PSO-BP model, and IPSO-BP model comparison tests, the IPSO-BPNN model improves the broadcast ephemeris orbit error correction rate of the MEO satellites in the range of 71.1–84%, and the broadcast ephemeris orbit error correction rate of IGSO satellites is improved by 70.2–80.2%, indicating that the IPSO-BPNN model has a better correction effect on satellites with different orbit types. Among the 23 MEO satellites, 12 satellites show the mean square error correction rates above 80% and the highest rate is above 84%. In this experiment, we do not exclude the existence of satellite noise data that interfere with the experiment and cause fluctuations in the accuracy, but the IPSO-BPNN broadcast ephemeris orbit correction model is more accurate than the BPNN and PSO-BPNN models overall. The correction rate is improved by 55.6–74.8% compared with the BPNN model and by 14.2–56.8% compared with the PSO-BPNN model. These studies showed that the optimization of the algorithm in this paper clearly produces improved results.

5. Conclusion

There is a certain error between the broadcast ephemeris orbit and the precise orbit of the satellite navigation system, and this error cannot be completely eliminated by the existing dynamical model. Through error analysis, it is found that there are some periodic regular phenomena in the broadcast ephemeris orbit errors. In this paper, the orbit error is modeled for this phenomenon using an improved particle swarm algorithm to optimize the BP neural network. Model improvement mainly uses nonlinear functions for inertia weights and learning factors to dynamically adjust the algorithm to accelerate its convergence while avoiding trapping in local optimal solutions. Experiments using BDS-3 satellite data show that the proposed method can construct an accurate orbit error model, and the accuracy of broadcast ephemeris orbit is significantly improved compared with BPNN and PSO-BPNN. In future research work, other deep learning methods, such as GRU and GAN, will be further combined with satellite navigation data processing to verify their effects on data accuracy improvement.

Data Availability

The satellite ephemeris data in this paper are provided by the iGMAS Wuhan station. The data can be found at the links below: (1) https://ftp://igs.gnsswhu.cn/pub/gps/data/daily/2022/. (2) https://ftp://igs.gnsswhu.cn/pub/gps/products/mgex/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the Natural Science Foundation of Shanxi Province, China (2021JQ-656).

References

[1] X. Xie, T. Geng, Q. Zhao, X. Liu, Q. Zhang, and J. Liu, "Design and validation of broadcast ephemeris for low Earth orbit satellites," *GPS Solutions*, vol. 22, no. 2, pp. 1–11, 2018.

[2] S. Pan, W. Chen, X. Jin, X. Shi, and F. He, "Real-time PPP based on the coupling estimation of clock bias and orbit error with broadcast ephemeris," *Sensors*, vol. 15, no. 7, pp. 17808–17826, 2015.

[3] M. Lihua and M. Wang, "Influence of ephemeris error on GPS single point positioning accuracy," *Artificial Satellites*, vol. 48, no. 3, pp. 125–139, 2013.

[4] M. Kim and J. Kim, "A long-term analysis of the GPS broadcast orbit and clock error variations," *Procedia Engineering*, vol. 99, pp. 654–658, 2015.

[5] Y. Youlong and L. Changjian, "Error analysis of GPS broadcast ephemeris a block IIF satellite perspective(in Chinese)," *Journal of Geodesy and Geodynamics*, vol. 36, no. 1, pp. 11–15, 2016.

[6] R. Yan, J. Liao, J. Yang, W. Sun, M. Nong, and F. Li, "Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering," *Expert Systems with Applications*, vol. 169, Article ID 114513, 2021.

[7] F. Wu, R. Jing, X.-P. Zhang, F. Wang, and Y. Bao, "A combined method of improved grey BP neural network and MEEMD-ARIMA for day-ahead wave energy forecast," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 4, pp. 2404–2412, 2021.

[8] B. Wu, L. Wang, and Y.-R. Zeng, "Interpretable wind speed prediction with multivariate time series and temporal fusion transformers," *Energy*, vol. 252, Article ID 123990, 2022.

[9] S.-X. Lv and L. Wang, "Deep learning combined wind speed forecasting with hybrid time series decomposition and multi-objective parameter optimization," *Applied Energy*, vol. 311, Article ID 118674, 2022.

[10] P. Yaqi, X. Chengdong, N. Fei, X. Zheng, and Y. Wang, "Prediction model of broadcast ephemeris orbit error based on PSO-BP neural network(in Chinese)," *Systems Engineering and Electronice*, vol. 41, no. 07, pp. 1617–1622, 2019.

[11] J. Changdong, Z. Jinshuai, and L. Hu, "LSTM algorithm for short-term prediction of BDS satellite clock bias," *Journal of Navigation and Positioning*, vol. 9, no. 1, pp. 44–52, 2021.

[12] H. Bohua, L. Xirui, X. Changfu, H. Ruofeng, H. Chongyan, and G. Zhiliang, "A CNN-based type recognition method for outliers of BDS satellite clock bias(in Chinese) [J]," *Geomatics and Information Science of Wuhan University*, vol. 46, no. 06, pp. 947–956, 2021.

[13] Y. Ye, H. Mo, and Y. Bin, "A high-precision medium-long term prediction method for navigation satellite clock bias(in Chinese)," *Chinese Journal of Scientific Instrument*, vol. 40, no. 09, pp. 36–43, 2019.

[14] X. Yu, J. Han, L. Shi, Y. Wang, and Y. Zhao, "Application of a BP neural network in predicting destroyed floor depth caused by underground pressure," *Environmental Earth Sciences*, vol. 76, no. 15, pp. 1–12, 2017.

[15] S. Jian, W. Lian-Guo, Z. Hua-Lei, and S. Yi-feng, "Application of fuzzy neural network in predicting the risk of rock burst," *Procedia Earth and Planetary Science*, vol. 1, no. 1, pp. 536–543, 2009.

[16] G. Gao, H. Zhang, H. San, X. Wu, and W. Wang, "Modeling and error compensation of robotic articulated arm coordinate measuring machines using BP neural network," *Complexity*, vol. 2017, Article ID 5156264, 8 pages, 2017.

[17] H. Chen, F. Niu, X. Su, T. Geng, Z. Liu, and Q. Li, "Initial results of modeling and improvement of BDS-2/GPS broadcast ephemeris satellite orbit based on BP and PSO-BP neural networks," *Remote Sensing*, vol. 13, no. 23, p. 4801, 2021.

[18] X. Chen, T. Wang, M. Ding, J. Wang, J. Chen, and J. X. Yan, "Analysis and prediction on the cutting process of constrained damping boring bars based on PSO-BP neural network model," *Journal of Vibroengineering*, vol. 19, no. 2, pp. 878–893, 2017.

[19] M. S. Arumugam and M. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing*, vol. 8, no. 1, pp. 324–336, 2008.

[20] X. Li, Y. Yuan, Y. Zhu et al., "Precise orbit determination for BDS3 experimental satellites using iGMAS and MGEX tracking networks," *Journal of Geodesy*, vol. 93, no. 1, pp. 103–117, 2019.

[21] W. Jiexian, W. Jungang, and C. Junping, "BDS broadcast ephemeris fitting based on satellite's and velocity(in Chinese) [J]," *Journal of Tongji University*, vol. 44, no. 1, pp. 155–160, 2016.

[22] W. Yan, S. Lijie, and H. Lingyong, "Analysis of perturbation forces of GPS precise orbit determination(in Chinese)," *Engineering of Surveying and Mapping*, vol. 22, no. 5, pp. 16–20+4, 2013.

[23] J. Wang, P. Shi, P. Jiang et al., "Application of BP neural network algorithm in traditional hydrological model for flood forecasting," *Water*, vol. 9, no. 1, p. 48, 2017.

[24] J. Zhou, X. Li, and X. Shi, "Long-term prediction model of rockburst in underground openings using heuristic algorithms and support vector machines," *Safety Science*, vol. 50, no. 4, pp. 629–644, 2012.

[25] H. Chaker, S. M. Meriah, and F. T. Bendimerad, "Optimization of micro strip array antennas using hybrid particle swarm optimizer with breeding and subpopulation for maximum side-lobe reduction," *Radioengineering*, vol. 17, no. 4, 2008.

[26] T. M. Khalil and A. V. Gorpinich, "Selective particle swarm optimization," *International Journal of Multidisciplinary Sciences and Engineering*, vol. 3, no. 4, 2012.

[27] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and its Applications," *Mathematical problems in engineering*, vol. 2015, Article ID 931256, 38 pages, 2015.

[28] W. Zhao, G. Liu, S. Wang, M. Gao, and D. Lv, "Real-time estimation of GPS-BDS inter-system biases: an improved particle swarm optimization algorithm," *Remote Sensing*, vol. 13, no. 16, p. 3214, 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1334081, 17 pages
https://doi.org/10.1155/2022/1334081

Research Article

A Multiobjective Optimization Model for a Dynamic and Sustainable Cellular Manufacturing System under Uncertainty

Javad Jafarzadeh,[1] Hossein Amoozad Khalili ,[2] and Naghi Shoja[3]

[1]*Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[2]*Department of Industrial Engineering, Sari Branch, Islamic Azad University, Sari, Iran*
[3]*Department of Mathematics, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran*

Correspondence should be addressed to Hossein Amoozad Khalili; amoozad92@yahoo.com

Received 7 May 2022; Accepted 28 June 2022; Published 22 September 2022

Academic Editor: Diego Oliva

For many years, cellular manufacturing has been implemented by owners of manufacturing units. Furthermore, the increasing importance of sustainable development has led manufacturers and managers to consider the concepts of sustainable manufacturing. Sustainable manufacturing includes three components: economic, environmental, and social responsibility. Many research and studies have been conducted in the field of cellular manufacturing, and also in most studies, only the economic component or at most two components of sustainable manufacturing have been taken into account. With increasing concerns about global warming, environmental issues have become particularly important in the production of products and goods. On the other hand, customer satisfaction as one of the aspects of social responsibility is of significant importance. In this research, we put the sustainable manufacturing system in the dynamic cellular manufacturing system under uncertainty (fuzzy parameters). A multiobjective sustainable mathematical model with objective functions of minimizing costs minimizing CO_2 emissions and minimizing product shortages (customer satisfaction) was proposed. In order to confirm the validity and accuracy of the proposed model, a small example was solved in GAMS software with CPLEX solver and epsilon constraint method, and its basic variables were investigated. Then, due to the high complexity of the proposed cellular manufacturing model, two meta-heuristic algorithms NSGA-II and MOGWO were used to solve larger problems in MATLAB software. To compare the performance of the two proposed algorithms, ten problems with different dimensions were designed and then the two algorithms were compared with each other based on several performance evaluation indicators. Also, in order to investigate the significance of the difference between the two algorithms based on each index, a statistical analysis was carried out by Minitab software. Taguchi method was also used to adjust the parameters of both algorithms. Based on the analytic results and statistical analysis, the MOGWO algorithm performed better than NSGA-II algorithm and the exact solution method, GAMS software.

1. Introduction

Dynamic cellular manufacturing is one of the newest manufacturing concepts that has been regarded in recent decades. The advantages of implementing it include in increasing the efficiency and productivity of workforce and space, reducing costs, reducing inventory while manufacturing and reducing costs, and improving manufacturing planning. Sustainable manufacturing has recently been considered by the owners of manufacturing firms. The benefits of its implementation include improving the morale and health of employees, employee responsibility and environmental protection, reducing pollution, and increasing profitability [1]. Considering the importance of the two mentioned issues, methods have been proposed to integrate the two.

The growing awareness of environmental protection has highly affected the production methods of manufactured products. Introduction of new methods demonstrates the features and advantages of sustainable manufacturing management over conventional and traditional methods. Sustainable manufacturing systems have attracted a great

deal of attention in the last 20 years (as an emerging manufacturing approach). Especially in the last 10 years, the number of papers focusing on the management of sustainable production systems has increased rapidly. There are more practical and integrated factors taken into account that make it more complex but closer to reality. Sustainable manufacturing is the creation of products produced through economic processes that minimize negative environmental impacts, while saving energy and natural resources, as well as increasing employee satisfaction, and community safety and product health [2].

Cellular manufacturing systems are among the new manufacturing methods that are used today in most of large manufacturing centers with high product diversity and multipurpose facilities. The basis of a cellular manufacturing system is the classification of products and machines based on their physical, operational, or processing similarity into several smaller manufacturing units called cells. Cells can be physical or virtual in nature. On the other hand, in most manufacturing units, the traditional perspective is used for supply and distribution planning. This means that each of these components independently plans for its activities. The supply chain in the manufacturing network seeks to create coordination between manufacturing facilities, suppliers, and allocation of products to product markets. Most manufacturing companies first design the supply chain and the number and locations of manufacturing facilities and make decisions about which product markets each facility serves. Then, they organize the processes (production line, cell arrangement, etc.) in each facility. In this case, the cellular arrangement will be done without considering the manufacturing network and chain. Simultaneously addressing the supply chain design and factory cellular manufacturing design will lead to the design of a manufacturing network with lower production and distribution costs to respond quickly to customers in return for having fewer facilities in manufacturing products. Furthermore, under real condition, the nature of many manufacturing parameters such as product demand, available machine capacity, and processing time are uncertain. Therefore, it is difficult to determine the cellular arrangement, the amount of manufacturing, and the amount of purchase according to the demand of customers' products in a competitive market that is facing uncertainty. Cellular manufacturing system is one of the new manufacturing methods that, in recent years, the industrial sector has benefited from its advantages, including reducing the amount and cost of transportation, reducing commissioning time, reducing manufacturing time, reducing package size, and reducing inventory during manufacturing.

Today, due to being in the movement of global competitive markets, manufacturing factories that are characterized by the specialization of needs, shortening of the product life cycle, shortening of the product supply cycle to the market and the various demands of customers should adapt to these conditions, and take measures that improve the efficiency and productivity of manufacturing processes [3]. Therefore, a mixed manufacturing system called dynamic cellular manufacturing was developed. This research has been conducted in order to achieve the main objective of integrating the problem of cell manufacturing and sustainable manufacturing by considering uncertainty in the safe scaffolding manufacturing industry and solving the proposed problem. Also, the subobjectives of the present study include considering environmental issues in the problem of cellular manufacturing, including the concept of social responsibility as one of the dimensions of sustainability in the problem of sustainable manufacturing and solving the problem of cellular manufacturing in conditions of uncertainty.

As mentioned, sustainable manufacturing has social, economic, and environmental aspects. The introduction of these concepts in industries and manufacturing will increase the efficiency and prosperity of manufacturing and respect for human issues psychologically and physically and respect for the environment in all its aspects and further progress in economic concepts. For example, in this study, an example of safe scaffolding manufacturing is given, which we do not enjoy its benefits in the country due to the lack of knowledge that is the manifestation of three criteria of sustainable manufacturing. Annually, a large number of workers lose their lives or become disabled due to noncompliance with safety issues. The urban environment is unbeautiful and unsafe because of old scaffolding. More iron is used to set up old scaffolding. This example alone illustrates the need for research in this area. And on the other hand, the benefits of cellular manufacturing mentioned earlier and considering both as integrated multiplies the importance of the issue. Even in industrialized countries, concerns about the issue of environmental pollution and greenhouse gases and other environmental issues and concerns are increasing every day. Social debates and not considering them in manufacturing decisions cause irreparable damage to society. All this increases the need for research on this subject.

Integration of dynamic cellular manufacturing systems by considering the concepts of sustainable manufacturing (economic, environmental, and social responsibility) in uncertainty is one of the innovations of this research. In this regard, we will try to provide multiobjective mathematical models to design a group arrangement in (1) formation of dynamic cells with the aim of decreasing the corresponding cost, (2) reduction in cost and time related to the concepts of sustainable manufacturing in the dynamic cellular manufacturing system and increasing the system efficiency in the real world in the manufacturing of safe scaffolding. Providing two meta-heuristic approaches in order to solving the proposed models, in this regard, we will also try to use meta-heuristic approaches in order to solve the proposed models. This research could be used in the future by researchers and students studying dynamic cellular manufacturing systems. One of the innovations of this research goes back to the fact that in addition to the usual research that is generally conducted in the field of dynamic cellular manufacturing systems, in this dissertation, sustainable manufacturing is also integrated with this issue, and three dimensions of sustainability (economic, environmental, and social responsibility) are used in this issue. Also, due to the consideration of uncertainty (fuzzy data), the

manufacturing environment is considered to be more realistic, and also meta-heuristic methods are used to solve the proposed problem. Furthermore, the implementation of this issue in the safe scaffolding manufacturing industry is another highlight of this research. Having an integrated view of the above points is an issue that is of great importance for the reasons mentioned. This is one of the research gaps (especially in the real world and especially scaffolding manufacturing). In other words, no research has been conducted in the country considering three dimensions of sustainable manufacturing simultaneously.

The rest of this research is organized in this way. In the second section, an introduction to the research literature and the history of the subject will be given. In the third section, while stating the proposed model, in order to confirm the validity and accuracy of the mathematical model, a problem with small dimensions in the safe scaffolding manufacturing industry is solved by the epsilon constraint method in GAMS software, and the basic variables are examined. In the fourth section, due to the NP-Hard nature of the proposed problem, two algorithms of NSGA-II and MOGWO are used to solve the mathematical model. Finally, conclusion and future suggestions are provided.

2. Literature Review

In this section, we describe the studies that have been carried out so far in the fields of cellular manufacturing, design of dynamic cellular manufacturing systems and cellular configuration and arrangement of cellular machines in dynamic conditions and sustainable manufacturing and (fuzzy) uncertainty.

Achieving a cellular manufacturing system requires three fundamental changes: a change in the arrangement and organization of machines and the formation of manufacturing cells, a change in the material ordering system and inventory control, and a change in the production planning system.

Saidi-Mehrabad and Safaei [4] proposed a comprehensive mathematical model with dynamic cellular configuration, alternative routing, part distribution, operation sequence, multiple units of identical machines, machine capacity, workload balance between cells, operation cost, subcontracting cost, cost of used tools, cost of preparation, size of cells, and proximity constraint of machines. Safaei and Tavakoli Moghaddam [5] assume a developed model in which parts in packets are moved between cells with assumptions such as plan of alternative operations, sequence of operations, capacity of machines, and capability multiplication of machines. The purpose of their model was to minimize the sum of the fixed and variable costs of machines, intercellular movement, and reconfiguration. Ah Kioon et al. [6] combined production planning and cellular manufacturing system in their research to minimize the total costs of intercellular material movement, storage of final items, preparation of the system for processing different parts in different time periods, and performing operations by machines.

Kia et al. [7] in their research described the problem of dynamic cellular arrangement using multiple conflicting goals. This paper presents the multiobjective nonlinear mathematical model on the dynamic cell formation (DCF) problem by weighting three conflicting objectives including the cost of moving machines in the cellular rearrangement process, the rate of utilization of machine capacity, and the total number of intracellular movements across the entire planning horizon. The model is solved by CPLEX, and because it is time consuming, the problem is solved using the scatter search algorithm. The results show the effectiveness of the scatter search algorithm. Rabbani et al. [8] considered the problem of designing cellular manufacturing systems in dynamic environments with changes in machine reliability and part demand. This paper proposes a multiobjective integer programming model for multiple planning horizons with the aim of maximizing machine reliability and minimizing system cost. The proposed model is solved using the branch and bound method and Lingo software.

Ahiska and Kurtul [9] in their study measured the performance of sustainable manufacturing with presence of recyclable waste, in which they used the two-stage data envelopment analysis method. Moldavska [10] expresses the evaluation of the model based on sustainable manufacturing. This paper examines the potential of studying a sustainable economy. The sustainable manufacturing model is based on the ideas of complexity theory. The development of the sustainable development index for the manufacturing industry is considered in this study; the definition of sustainability is clear: manufacturing is positive and at the same time has little effect on inflexible resources. The purpose of this study is to determine the sustainability index of the manufacturing plant. Since the topic is so wide-ranging, this research is limited to small and medium-sized industries with defined set of joint operations and process plans.

Ma et al. [11] reviewed more than 100 related papers, mainly 1994–2015, in the field of classification of mathematical problems with the management of sustainable manufacturing systems and divided them into three categories according to the main elements in a manufacturing system: production planning and control, inventory management, and control and building of network design. A major challenge for manufacturers is not only designing but also producing products using a sustainable approach. Sustainable development consists of three structural pillars: society, environment, and economy but meanwhile includes operational aspects such as resource consumption, natural environment, economic performance, workers, products, social justice, and society development. Manufacturing industries have recognized that their task is to design a sustainable manufacturing system that has fewer environmental impacts and social disruptions and promotes wealth. One of the key results in this study is that technology capability and economic risk are the two main factors that prevent a company from making a decision to implement sustainable manufacturing.

Singh et al. [12] in their study provided a specialized system based on fuzzy law to evaluate the sustainable performance of manufacturing in small and medium

enterprises. The initial set of measures and measurements were determined based on their characteristics. Sixteen criteria were identified and classified under four economic measures, five environmental measures, and three social measures.

Koren et al. [13] stated 6R (reduction, reuse, recycle, recover, redesign, and remanufacturing) in their research and considered it as a basis for the factory to manufacture the next generation, which can significantly increase system sustainability, quickly adjust system configuration and production processes to respond to emerging market demand, and maintain system values for future products. Mutingi et al. [14] used the modular production design approach for sustainable manufacturing in the fuzzy environment. Salido et al. [15] stated in their research that manufacturing industries face environmental challenges; therefore, their industrial processes should be optimized in terms of profitability and sustainability. Since many of these processes are dynamic, this paper addresses the improvement in dynamic planning problems in workshop manufacturing where machines can operate at different rates. The results provided in this paper may be useful for use in real industries to redesign cost-effective manufacturing.

Singh and Sultan [16] in their research provided a model using the theory of complexity to evaluate sustainability in sustainable manufacturing. Fisher et al. [17] introduce research in the field of cloud manufacturing as a future for application in sustainable manufacturing. In their research, they introduced four methods to increase sustainability, including calibrating the design and increasing automation.

Peng et al. [18] provide a paper to improve the manufacturing sustainability of workshop machinery in workshop manufacturing. This paper provides a common multiobjective model of energy consumption and production efficiency. Moldavska and Martinsen [19] stated in their research that the number of scientific papers in the field of sustainable manufacturing shows a significant growth in interest in this subject in the last 20 years. Contrary to many published papers, the deep purpose of sustainable manufacturing or at least a support of a strong theory is still missing. 6R is the first attempt to resolve this issue. In this study, they sought to expand this concept.

Malek and Desai [20] stated that sustainable manufacturing is a combination of the triple bottom approach (economic, environmental, and social) from manufacturing methods. In such a complex system, decision-making becomes difficult in terms of selecting and prioritizing various aspects. Multiple multicriteria decision-making methods can facilitate selection and prioritization in a complex system. Their research prioritizes sustainable manufacturing barriers by calculating weight by using the best and worst methods in an Indian manufacturing organization. Beekaroo et al. [21] in a study entitled development of a sustainability index for Mauritian manufacturing companies examined the sustainability indices in 30 Mauritian companies in terms of nine environmental indices, four economic indices and two social indices.

Zheng et al. [22] in their research addressing to energy saving and pollution reduction by reducing production deficits. While previous research has focused on reducing time and labor. Pagone et al. [23] discuss the concept of decision-making in the selection of parts according to sustainability criteria, in which 18 criteria are divided into four groups of time, cost, quality, and environment, and through TOPSIS, they concluded that aluminum alloy would be the best choice for manufacturing automobile parts.

Pourghader Chobar et al. [24] proposed a multiobjective location-routing problem model for multidevice relief logistics under uncertainty using meta-heuristic algorithm. Rezaei Kallaj et al. [25] presented a research in which the problem of vehicle routing in relief supply under a crisis condition considering blood supply has been presented. Pourghader Chobar et al. [26] proposed a multiobjective model for hub location problem considering dynamic demand and environmental issues.

Table 1 summarizes the comparison of research conducted regarding the subject literature, based on which research gaps are explored.

In order to take advantage of the benefits of dynamic cellular manufacturing and sustainable manufacturing, which were briefly mentioned above, we need to consider these two concepts integrated. The research gap that is the reason for conducting this research is also this issue that as mentioned in the tables above has received little attention. In this research, we put the dynamic and sustainable cellular manufacturing system in conditions of uncertainty. Operators have not been considered by researchers in previous research, while they are the basis for implementing sustainable manufacturing. The main aspects of sustainability are economic, environmental, and social responsibility, which in this study are considered as objective functions of the proposed mathematical model.

3. Problem Definition and Mathematical Modeling

In this section, according to the research gaps mentioned in the previous section, the place of the forthcoming research was determined, followed by the definition of the problem and the description of the research assumptions in such a way that it can cover the shortcomings of the study area as much as possible. First, the fuzzy mathematical model is provided and then its defuzzification method is explained, and the defuzzificated model is provided. Finally, an example in small dimensions is solved in GAMS software by CPLEX solver and epsilon constraint method to validate the proposed model and the results are expressed.

Cellular manufacturing systems are among the new manufacturing methods that are used today in most of large manufacturing centers with high product diversity and multipurpose facilities. The basis of a cellular manufacturing system is the classification of products and machines based on their physical, operational, or processing similarity into several smaller manufacturing units called cells. Cells can be physical or virtual in nature. On the other hand, in most manufacturing units, the traditional perspective is used for supply and distribution planning. This means that each of these components independently plans for its activities. Today, due to being in the movement of global competitive

Table 1: Literature review of sustainable manufacturing dimensions.

Social dimension	Economy	Salary and benefits	[12]
	Satisfaction level	Level of employee satisfaction, customer churn	[6]
	Quality and health	Employee health and safety programs, human factor engineering, average travel distance of employees to the company	[18] [7]
	Human resources and society	Workforce access, workforce skills, workforce selection, employee learning hours, employee performance appraisal, society development	[11] [12]
Environmental dimension	Environmental management	Environmental standards, environmental characteristics and goals, structure responsible for the environment, environmental assessment	[7] [8] [24]
	Environmental aspects, responsibility	Relationship between suppliers and the environment, the company's view of the environment, the type of waste treatment, the consumption of hazardous substances	[17] [1]
	Consumption	Water, energy, and paper	[23]
Economic dimension	Economic management	Market distribution, number of recycled materials	[22]
	Operational efficiency and products	Life cycle, lean manufacturing waste, start-up time, flexibility, inventory, quality of products and services, society quality management, new products, initiative in international markets, design for production and assembly	[13]
	Operational results and suppliers	Profit, price and operational indicators, standard for suppliers, delivery	[27] [8]
	Customers	Number of customer complaints about the area, delivery time	[22] [28]
	Infrastructure	Proximity to transportation centers, hubs, access to alternative transportation, access to storage facilities, efficiency of using transportation resources, access to manufacturing facilities	[17] [1] [8]

markets, manufacturing factories that are characterized by the specialization of needs, shortening of the product life cycle, shortening of the product supply cycle to the market, and the various demands of customers should adapt to these conditions and take measures that improve the efficiency and productivity of manufacturing processes. Therefore, a mixed manufacturing system called dynamic cellular manufacturing was developed. Sustainable manufacturing has economic, social, and environmental dimensions. Sustainable manufacturing is the creation of products produced through economic processes that minimize negative environmental impacts, while saving energy and natural resources, as well as increasing employee satisfaction, and society safety and product health.

3.1. Mathematical Modeling. The assumptions of the problem are as follows:

(1) Shortage is permissible and limited.

(2) Each machine has a fixed cost as an overhead cost and is not related to its consumption.

(3) The variable cost of each machine depends on its working hours.

(4) If a machine is purchased, it must be in the workshop in the next time periods and the removal of the machine is not considered.

(5) Partitioning between cells is not considered and one place can be assigned to different cells.

(6) Moving machines includes installation and removal. When a machine is moved from one place to another, both installation and removal costs are covered. Also, regardless of the cost of purchasing a machine, the installation cost is considered for the machines.

(7) Intracellular and extracellular transmission costs of products are linear function of distance.

(8) For the number of machines assigned to each cell, an upper and lower limit is considered. The large number of machines in each cell complicates the production control in the cell. On the other hand, the small number of machines in the cell increases the intercellular transmission costs.

(9) The shape and position of the cells are not predetermined.

(10) Operators based on their expertise can operate on the machines in question and there is no limit to the number of operators.

In this section, the desired fuzzy mathematical model is proposed. The proposed mathematical model has three objective functions. The constraints and objective functions and a brief description of them are provided below.

3.1.1. Sets

p: set of product types $p\{1, 2, \ldots, P\}$

t: set of time periods $t\{1, 2, \ldots, T\}$

r: set of operations required to perform on products $r\{1, 2, \ldots, R_p\}$

$c, \acute{c}$: set of cells $c\{1, 2, \ldots, C\}$

$l, \acute{l}$: set of places for establishment of machines $l\{1, 2, 2 \ldots, L\}$

$w, \acute{w}$: set of operators based on expertise $w\{1, 2, \ldots, W\}$

3.1.2. Parameters

$\widetilde{CI}_p$: intracellular transmission cost per unit of product p per unit of distance

$\widetilde{CE}_p$: intercellular transmission cost per unit of product p per unit of distance

$\widetilde{d}_{pt}$: demand for product p at time t

Q_{ll}: distance between places l and l'

$\widetilde{CINS}_m$: installation cost of machine type m

$\widetilde{CPUT}_m$: cost of removing machine type m

$\widetilde{CF}_m$: overhead cost of machine type m in each time period

CV_m: variable costs of machine type m

CB_m: cost of purchasing machine type m

$Ucap_c$: maximum number of machines to be assigned to cell c.

$Lcap_c$: minimum number of machines to be assigned to cell c.

A_m: area of machine type m

$\widetilde{\omega}_{prm}$: processing time of the rth operation for product p on machine type m

$\widetilde{cap}_m$: time capacity of machine type m in each time period

$\widetilde{Tcap}$: total capacity of the workshop to store products

$Gcap$: maximum limit for carbon dioxide emissions

V_p: volume of product p

$\widetilde{CO}_w$: cost of each working day of the operator with expertise in w

g_{pm}: Co_2 emission rate due to the production of each unit of product p on machine type m

$\widetilde{CH}_p$: inventory cost per unit of product p

SS_p: maximum shortage product of p in each period

3.1.3. Decision Variables

x_{clmwt}: if the operator with expertise w is assigned to machine type m in place l in cell c at time t it is 1, otherwise 0.

y_{prmw}: if the rth operation of product p can be performed on machine type m by the operator with expertise w it is 1, otherwise 0.

S_{pt}: shortage of product p at time t

I_{pt}: amount of product p at time t

$u_{prclmtw}$: number of p-type products on which the rth operation was performed by machine type m in place l in cell c at time t by the operator with expertise w.

$NP_{prc\acute{c}l\acute{l}m\acute{m}\acute{n}t}$: number of p-type products that were processed at time t by the rth operation on machine type m in place l in cell c, and now they were transferred on machine m' in cell c' in place l' to perform the $r+1$th operation.

v_{lmt}: if place l is empty at time $t-1$ or assigned to a machine other than machine m and assigned to machine m at time t it is 1, otherwise 0.

d_{lmt}: if machine type m is assigned to place l at time $t-1$ and removed at time t or replaced by another machine it is 1, otherwise 0.

N_{mt}: total number of m-type machines that are added to the workshop level at the beginning of period t.

NO_{wt}: number of operators available with expertise w at the workshop level at time t.

Before providing the mathematical model of the problem, since some of the parameters of the problem are considered uncertain, we introduce the possibilistic programming model and then express the final uncertain mathematical model that is modeled using the linear possibilistic programming.

Philosophically, fuzzy logic expresses a concept of logic in which, unlike Aristotelian logic, which rules classical mathematics, it uses multivalued logic to express its concepts. This logic, unlike Aristotelian logic, which always considers everything as being or not being, does not assume a boundary between being and not being. In this logic, anything can belong to a group to a certain extent. The important point from our point of view in this dissertation is the application of fuzzy models on the subject of this dissertation. The complex nature of DSCM makes its mathematical model parameters uncertain. As mentioned, there are several types of uncertainty, one of which is possibilistic programming uncertainty, which can be cited when there is insufficient information about the parameters. And since the parameters and information about the DCMS problem are uncertain, fuzzy logic is used for the input information. The research method of [29] is used to convert the possibilistic model of the problem, which includes inaccurate coefficients in both the objective function and the constraints, into a certain model. This method is computationally efficient because it retains the linearity property and also does not increase the number of objective functions and unequal constraints. Due to the simplicity of obtaining the data, the triangular fuzzy distribution is used to model the inaccurate nature of the ambiguous parameters of the problem. Assume that $\widetilde{c} = (c^{(1)}, c^{(2)}, c^{(3)})$ is a triangular fuzzy number and the following model is considered:

$$\begin{aligned} \text{Min } Z &= cx \\ \text{s.t: } Ax &\le b \\ Dx &\ge f \\ x &\ge 0. \end{aligned} \tag{1}$$

Now, assuming that x is the decision variable and c, b, A, D, and f are the model parameters and the parameters c, b, and f are fuzzy, then the defuzzificated model is as follows:

$$\begin{aligned}\text{Min } Z &= \left[\frac{c^{(1)} + c^{(2)} + c^{(3)}}{3}\right]x \\ \text{s.t: } Ax &\le \left[(2\alpha - 1)b^{(1)} + (2 - 2\alpha)b^{(2)}\right] \\ Dx &\ge \left[(2\alpha - 1)f^{(3)} + (2 - 2\alpha)f^{(2)}\right] \\ x &\ge 0.\end{aligned} \tag{2}$$

Defuzzification is from the Fazle method. C, b, and f are uncertain and have a triangular membership function, in the objective function, the average of the parameter is uncertain and in the constraint of greater than or equal to like demand, the criterion for satisfying demand is the larger number, and in the constraint of smaller than or equal to like capacity, the criterion is the smaller number. Just before each fuzzy parameter in the constraint proportional to the larger and smaller inequality, according to Relation (2), there is an expression that by changing the coefficient α, which is actually the level of confidence, sensitivity analysis is carried out on fuzzy parameters. The higher this coefficient becomes and gets closer from 0.5 to 1, the more number is multiplied by the pessimistic number and the closer it gets to 0.5, the more value is multiplied by the possible number.

The parameter α must take a value in the range of 0.5 to 1. Based on the method of [29].

The final uncertain model:

$$\begin{aligned}\text{Min } Z_1 = &\sum_{t,p,r,c,\acute{c},l,\acute{l},m,\acute{m}} U_{prc\acute{c}l\acute{l}m\acute{m}w\acute{w}t} \times Q_{l\acute{l}} \times \left[\frac{CE_p^1 + CE_p^2 + CE_p^3}{3}\right] \\ &+ \sum_{t,p,r,c,l,\acute{l},m,\acute{m}} NP_{prc\acute{c}l\acute{l}m\acute{m}w\acute{w}t} \times Q_{l\acute{l}} \times \left[\frac{CI_p^1 + CI_p^2 + CI_p^3}{3}\right] \\ &+ \sum_{t=2,l,m} v_{lmt} \times \left[\frac{CINS_m^1 + CINS_m^2 + CINS_m^3}{3}\right] \\ &+ \sum_{t=2,l,m} d_{lmt} \times \left[\frac{CPUT_m^1 + CPUT_m^2 + CPUT_m^3}{3}\right] + \sum_{t,c,l,m,w} x_{clmwt} \times \left[\frac{CF_m^1 + CF_m^2 + CF_m^3}{3}\right] \\ &+ \sum_{t,p,r,c,l,m,w=1} u_{prclmwt} \times \left[\frac{\omega_{prm}^1 + \omega_{prm}^2 + \omega_{prm}^3}{3}\right] \times \left[\frac{CV_m^1 + CV_m^2 + CV_m^3}{3}\right] + \\ &\sum_{t,m} N_{mt} \times CB_m + \sum_{t=2,l,m} x_{clmwt} \times \left[\frac{CINS_m^1 + CINS_m^2 + CINS_m^3}{3}\right] \\ &+ \sum_{t,p} I_{pt} \times \left[\frac{CH_p^1 + CH_p^2 + CH_p^3}{3}\right] + \sum_{w,t} NO_{wt} \times \left[\frac{CO_w^1 + CO_w^2 + CO_w^3}{3}\right],\end{aligned} \tag{3}$$

$$\text{Min } Z_2 = \sum_{t,p,r,c,l,m,w} u_{prclmwt} \times g_{pm}, \tag{4}$$

$$\text{Min } Z_3 = \sum_{p,t} S_{pt}, \tag{5}$$

$$\begin{aligned}&u_{prclmwt} \le y_{prmw} * x_{clmwt} * \left[(2\alpha - 1)d_{pt}^{(1)} + (2 - 2\alpha)d_{pt}^{(2)}\right] \\ &\forall p, r = \left\{1, \ldots, R_p\right\}, c, l, m, w, t,\end{aligned} \tag{6}$$

$$\sum_{c,l,m,w} u_{prclmwt} + I_{p,t-1} + S_{pt} - I_{pt} = \left[(2\alpha - 1)d_{pt}^{(3)} + (2 - 2\alpha)d_{pt}^{(2)}\right] \quad \forall p,\ r = R_p, t, \tag{7}$$

$$\sum_{l,m,w} x_{clmwt} \times A_m \le Ucap_c \forall c, t, \tag{8}$$

$$\sum_{l,m,w} x_{clmwt} \times A_m \le Ucap_c \forall c, t. \tag{9}$$

$$\sum_{l,m,w} x_{clmwt} \times A_m \geq Lcap_c \forall c, t, \tag{10}$$

$$\sum_{p,r,c,w} u_{prclmwt} \times \omega_{prm} \leq \left[(2\alpha - 1)\text{cap}_m^{(1)} + (2 - 2\alpha)\text{cap}_m^{(2)}\right] \forall l, m, t, \tag{11}$$

$$\sum_{c,l,m,w} NP_{prc\acute{c}l\acute{l}m\acute{m}t} = u_{p,r+1,c\acute{l}\acute{m}\acute{w}t} \quad \forall p, \acute{c}, \acute{l}, \acute{m}, \acute{w}, t, r = 1, \ldots, R_p - 1, \tag{12}$$

$$\sum_{\acute{l},\acute{c},\acute{m},\acute{w}} NP_{prc\acute{c}\acute{l}\acute{l}m\acute{m}t} = u_{prclmwt} \quad \forall p, c, l, m, w, t, r, \tag{13}$$

$$\sum_{c,l,w} x_{clmw,t+1} - \sum_{c,l,w} x_{clmwt} = N_{m,t+1} \quad \forall m, t = 1, \ldots, T - 1\, v, \tag{14}$$

$$\sum_{c,l,w} x_{clmw,1} = N_{m,1} \quad \forall m, \tag{15}$$

$$\sum_{c,l,w} x_{clmw,1} = NO_{wt} \forall w, t, \tag{16}$$

$$\left(1 - \sum_{c,w} x_{clmwt}\right) \times \sum_{c,w} x_{clmw,t+1} = v_{lm,t+1} \quad \forall l, m, t = 1, \ldots, T - 1, \tag{17}$$

$$\sum_{c,w} x_{clmwt} \times \left(1 - \sum_{c,w} x_{clmwt,t+1}\right) = d_{lm,t+1} \quad \forall l, m, t = 1, \ldots, T - 1, \tag{18}$$

$$\sum_{p} I_{pt} \times V_p \leq \left[(2\alpha - 1)T\text{cap}^{(1)} + (2 - 2\alpha)T\text{cap}^{(2)}\right] \forall t, \tag{19}$$

$$\sum_{p,r,c,l,m,w} u_{prclmwt} \times g_{pm} \leq G\text{cap} \quad \forall t, \tag{20}$$

$$I_{p,0} = 0 \quad \forall p, \tag{21}$$

$$\sum_{p} S_{pt} \leq SS \quad \forall t, \tag{22}$$

$$\begin{aligned} &d_{lmt}, v_{lmt}, y_{prmw}, x_{clmwt} = \{0, 1\}, \\ &S_{pt}, I_{pt}, u_{prclmtw}, NP_{prc\acute{c}\acute{l}\acute{l}m\acute{m}t}, N_{mt}, NO_{wt} \geq 0. \end{aligned} \tag{23}$$

Relation (3) is the first objective function that is to minimize costs. In fact, it is considered as an economic objective function that each component is the cost of transferring products between cells, the cost of transferring products inside cells, the cost of installing machines, the cost of removing machines, the overhead cost of machines, the variable cost of machines per operating time, the cost of purchasing machines, the cost of installing machinery in the first period, the cost of maintaining products in the workshop warehouse, and also the last component is the cost of the operator per period of time, respectively. Relation (4) is the second objective function that is to minimize the amount of carbon dioxide emissions by each type of machine per unit of product manufacturing. It is also introduced as an environmental objective function. Relation (5) as the third objective function is to minimize the shortage of estimation of customer demand, which conveys the concept of increasing customer satisfaction, and is called the objective function of social responsibility.

Relation (6) states that products can be manufactured by the relevant machines and operators to the maximum of the demand in each period. There must also be the ability to manufacture the desired product by the machine and the operator assigned. Relation (7) shows the inventory level of the workshop warehouse as well as the lost opportunity to supply the product (shortage) at any time for each type of

product. In fact, it states that the amount of product manufactured in each time period plus the shortage and inventory of the previous period must be equal to the amount of demand plus inventory of this period. Relation (8) shows that each place in each time period can be assigned to a maximum of one machine. Relation (9) shows the maximum capacity of the cell for the placement of machines. Relation (10) shows the minimum number of machines to be placed in each given cell. Relation (11) indicates the time capacity of each machine in each time period to manufacture a product. In other words, each product needs an amount of time to manufacture, and this relation indicates the allowable amount of using each machine in each time period. Relations (12) and (13) are the constraints of maintaining the flow. Relation (14) shows the number of machines added to the workshop level in time periods. In fact, it shows the number of machines purchased in each period. Relation (15) shows the number of machines purchased in the first time period. Relation (16) shows the number of operators in time periods. Relations (17) and (18) show the amount of movement of machines or, in fact, the installation and removal of machines in each time period. Relation (19) shows the warehouse capacity of the workshop to store products after manufacturing. Relation (20) states the allowable amount of Co_2 emissions. Relation (21) indicates that the warehouse is empty on day zero. Relation (22) states that the shortages must not exceed a certain amount.

It should be noted that the constraints related to Relations (17) and (18) are nonlinear and are linearized as follows (Bayram and Sahin [30]):

$$\begin{aligned}
&0.5 + v_{lm,t+1} + \sum_{c,w} x_{clmwt} - \sum_{c,w} x_{clmw,t+1} \geq 0 \quad \forall l, m, t = 1, \ldots, T-1,\\
&1.5 + v_{lm,t+1} + \sum_{c,w} x_{clmw,t+1} - \sum_{c,w} x_{clmw,t+1} - 1 \leq 0 \quad \forall l, m, t = 1, \ldots, T-1,\\
&0.5 + d_{lm,t+1} + \sum_{c,w} x_{clmwt} - \sum_{c,w} x_{clmwt} \geq 0 \quad \forall l, m, t = 1, \ldots, T-1,\\
&1.5 + d_{lm,t+1} + \sum_{c,w} x_{clmwt} - \sum_{c,w} x_{clmw,t+1} - 1 \leq 0 \quad \forall l, m, t = 1, \ldots, T-1.
\end{aligned} \tag{24}$$

There is no variable change in the linearization performed by the base paper, and the linearization is performed as presented above.

4. Solving the Problem and Providing the Results

4.1. Validation of the Proposed Model. In order to validate the proposed model, an example with small dimensions and its decision variables are examined to confirm the accuracy of the model. Due to the multipurpose nature of the proposed model, the augmented epsilon constraint method in GAMS software is used to provide the Pareto front related to the three objective functions of this model. The number of product types is 2, the number of operations is 3, the number of cells is 2, the number of places for machines is 5, the number of types of machines is 2, and the type of expertise of operators is 3. And this manufacturing of products is done in 4 days. The results of solving the proposed problem by GAMS software and the AEC method and the Pareto front formed are shown in the next steps.

Based on the two-dimensional Pareto front formed by the AEC method, it can be concluded that each of the two and three objective functions is in conflict with the first objective function, meaning that with increasing costs separately, carbon dioxide emissions are reduced, and the shortages of meeting the demand are also decreased. In this section, the optimal point for analysis is selected as follows: (cost = 567451, environment = 23014, social responsibility = 172). Also, GAMS software can solve problems 1 to 4 (Table 2), and subsequent problems are almost unsolvable due to a significant increase in solution time, and this issue, as well as the expression of the complexity of this problem by other authors is proof that this problem is NP-Hard.

4.2. Solving the Problem with Meta-Heuristic Algorithms. After solving the problem by GAMS software, due to the NP-Hard nature of the proposed model, meta-heuristic algorithms are used. NSGA-II and MOGWO algorithms are used to solve this model. Then, to compare the two algorithms, 10 problems are designed, and each problem is executed by the algorithms 10 times. The algorithms are compared based on 5 indicators. First, the problem solved in the previous chapter is solved by the two proposed algorithms, and the Pareto front of all three methods is shown. It is then represented by αs with different values of change in objective functions. Then, the two algorithms are compared with each other by SNS, RAS, MID, Time, and Diversity metrics, and the results are presented by statistical analysis in Minitab software. To compare the algorithms based on 5 metrics: SNS, RAS, Diversity, MID, and Time, 10 problems are designed; the problem information is presented in Table 3. It should be noted that GAMS software has the ability to solve problems 1 to 4, and in solving the next problems, the solution time increases significantly, and they cannot be solved. Both this issue and the fact that other papers have found this problem to be complex is proof that this problem is NP-Hard.

Table 2: EC output from GAMS.

Problem	EC					
	Time	Diversity	Spacing	Mid	SNS	RAS
3 ∗ 2 ∗ 2 ∗ 2 ∗ 2 ∗ 2 ∗ 2 ∗ 2	23.256	897225	7411	214558	147226	0.24755
3 ∗ 2 ∗ 2 ∗ 3 ∗ 2 ∗ 2 ∗ 2	88.256	1036685	11478	474558	478226	0.14778
3 ∗ 3 ∗ 3 ∗ 4 ∗ 3 ∗ 3 ∗ 3	178.669	1255369	18996	741058	1455230	0.01477
3 ∗ 3 ∗ 3 ∗ 5 ∗ 3 ∗ 3 ∗ 3	879.614	2476698	29880	1011475	1854701	0.08958

Table 3: Information on 10 numerical problems.

Problem dimensions ($P * R * C * L * M * W * H$)	Uniform data	Parameter	Uniform data	Parameter
3 ∗ 2 ∗ 2 ∗ 2 ∗ 2 ∗ 2 ∗ 2 ∗ 2	[5, 7]	ω_{prm}	[70, 50]	d_{ph}
3 ∗ 2 ∗ 2 ∗ 3 ∗ 2 ∗ 2 ∗ 2	[750, 650]	cap_m	[80, 60]	I_p
3 ∗ 3 ∗ 3 ∗ 4 ∗ 3 ∗ 3 ∗ 3	[1, 0]	δ_{prmw}	[150, 110]	E_p
3 ∗ 3 ∗ 3 ∗ 5 ∗ 3 ∗ 3 ∗ 3	[1300, 1100]	$T\mathrm{cap}$	[60, 40]	$Q_{ll'}$
4 ∗ 4 ∗ 4 ∗ 6 ∗ 4 ∗ 4 ∗ 4	[1800, 1600]	$G\mathrm{cap}$	[500, 450]	μ_m
5 ∗ 5 ∗ 5 ∗ 7 ∗ 5 ∗ 5 ∗ 5	[17, 12]	V_p	[300, 250]	π_m
5 ∗ 5 ∗ 5 ∗ 8 ∗ 5 ∗ 5 ∗ 5	[160, 120]	OC_w	[110, 70]	FC_m
6 ∗ 6 ∗ 6 ∗ 10 ∗ 6 ∗ 6 ∗ 6	[70, 30]	g_{pm}	[70, 30]	VC_m
6 ∗ 6 ∗ 6 ∗ 12 ∗ 6 ∗ 6 ∗ 6	[50, 30]	HC_p	[450, 270]	φ_m
7 ∗ 7 ∗ 7 ∗ 15 ∗ 7 ∗ 7 ∗ 7	[3, 2]	A_m	[6, 5]	$U\mathrm{cap}_c$
			[4, 2]	$L\mathrm{cap}_c$

The problem is solved by the two proposed algorithms, and the Pareto front formed is shown in Figure 1. Then, a sensitivity analysis on α is carried out on the objective functions. The results are presented in Figure 2.

According to Figure 2, the algorithms have generated a Pareto front close to the Pareto front of the exact method, which is the augmented epsilon constraint (AEC). For each answer to the problem, this string is mapped to the problem variables. The output of the gray wolf algorithm in Table 4.

The output of the genetic algorithm is also Table 5.

As shown in Figure 3, with increasing α, due to the increasing effect on the pessimistic value of fuzzy parameters and the nature of minimizing the objective functions of these functions, their values increase. This means that with increasing demand for products and increasing costs of installation and purchase of machinery and fixed and variable manufacturing costs, costs, carbon dioxide emissions, and shortages increase.

4.3. Parameter Adjustment the by Taguchi Method. In Taguchi method, the factors affecting the test result are divided into two categories: uncontrollable (so-called noise (N)) and controllable (so-called signal (S)). An S/N variable is then defined, which is the signal-to-noise ratio. Taguchi parameter setting method adjusts the factors to levels that maximize the S/N ratio. One of the meta-heuristic methods used in this research is NSGA-II algorithm, four parameters MaxIt, NPOP, PC, PM must be set at optimal levels. For this purpose, first, three levels of low (1), medium (2), and high (3) are defined separately for each parameter to solve the problems, which are given in Table 6. Then, the set of experiments proposed by Taguchi method is calculated for four factors in three levels, in which 9 different modes are designed by Taguchi method. (It should be noted that each experiment is performed 10 times and their average is recorded, which is done to reduce the error of the algorithms and the answer will be more reliable.)

Now, in order to create an output from each experiment using the following method, all the indicators will be converted into one answer. First, the nature of each indicator based on negativity (less - better) or positivity (more - better) must be determined, in which case, the more the Diversity index, the better, and other indicators are negative in nature. Table 7 shows the different indicators and positions that are the same designed experiments defined by O_i.where x_j and Q_i represent the j-th criterion and the i-th position, respectively. Also, r_{ij} is equal to the value that position i in the indicator j will take. The obtained values of the indicators are normalized according to the fuzzy unscaling technique in Relations.

$$x_j^+ \longrightarrow R_{ij} = \frac{r_{ij} - \min\left(r_{ij}\right)}{\max\left(r_{ij}\right) - \min\left(r_{ij}\right)}, \tag{25}$$

$$x_j^- \longrightarrow R_{ij} = \frac{\max\left(r_{ij}\right) - r_{ij}}{\max\left(r_{ij}\right) - \min\left(r_{ij}\right)}. \tag{26}$$

In this method of normalization, indicators with a negative nature will have a positive nature. Using the goal programming approach, the indicators are prioritized in terms of importance and the weight for each indicator is considered accordingly. In this research, the importance of each weight is applied according to Rabbani et al. [8]. Then, according to the significance coefficients, the total weight of the indicators of each experiment is calculated in the *Response* equation.

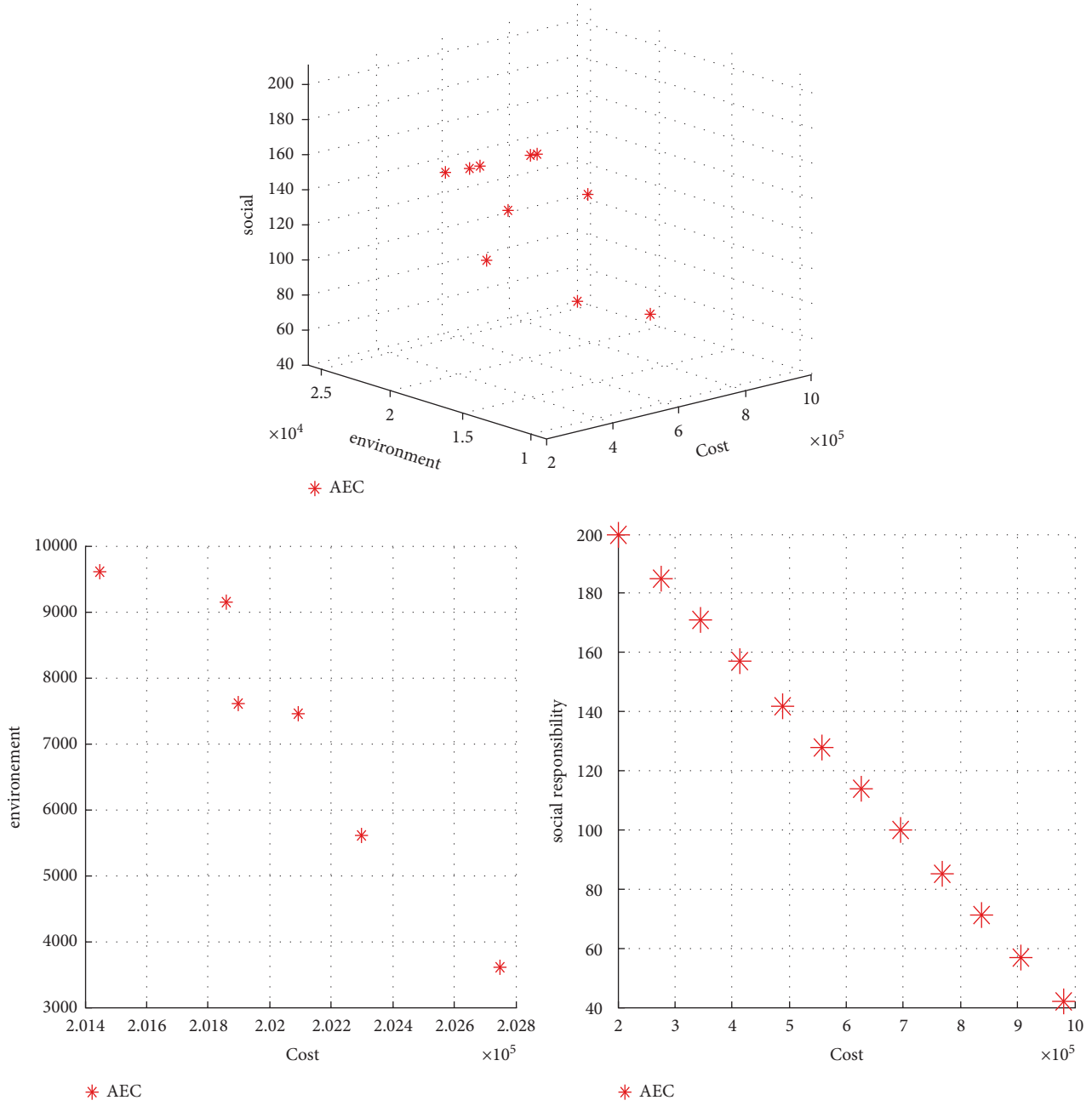


Figure 1: Three-dimensional and two-dimensional Pareto fronts resulting from problem solving for model validation.

$$\text{Response} = \sqrt[2]{(MI\ D)^2 + (RAS)^2 + (SNS)^1 + (SNS)^1 + (\text{Diversity})^1 + (\text{Spacing})^1}. \tag{27}$$

Now, based on the response values of each algorithm, the S/N rate is calculated and based on it, the levels of each input parameter are determined. To do this, the resulting response is presented as the final response to the experiments, and the Taguchi more-better formula is used to calculate the S/N ratio. The results of the algorithms are given in Tables 8–12, respectively.

4.4. Evaluation Indicators. If a single-objective optimization problem Z_1 minimization problem) is considered, it is clear that any solution that offers a possible solution for which Z_1 is less is better. But in the case of the multiobjective decision-making (MODM) problem, the evaluation method is different, and it cannot be evaluated as single objective. In general, in evaluating MODM solution methods, it is

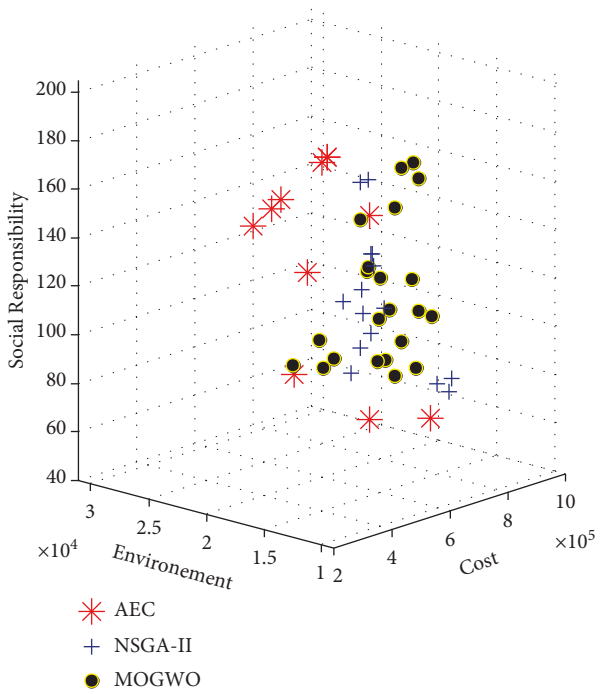


Figure 2: Pareto front formed by two methods of solving the small problem.

Table 4: Output of gray wolf algorithm.

The first objective function	174580.39
The second objective function	4750
Third objective function	257
Sum $u_{prclmhw}$	145
Sum $U_{prcc'll'mmh'}$	65
Sum of deficiency	245
Deficit rate S_{ph}	11 13 5 6 0 8 14 4 5 0
Inventory amount I_{ph}	0 15 5 17 0 0 26 25 14 0

Table 5: Output of the genetic algorithm.

The first objective function	165227.58
The second objective function	5620
Third objective function	320
Sum $u_{prclmhw}$	92
Sum $U_{prcc'll'mmh'}$	65
Sum of deficiency	285
Deficit rate S_{ph}	8 20 15 1 0 4 5 7 1 0
Inventory amount I_{ph}	0 16 14 1 0 0 25 12 25 1

important to note that each method has better performance with the criterion of greater maximum diversity, the criterion of less distance from the ideal solution (MID), the less value of RAS and the criterion of spacing, and more SNS. Accordingly, we define the following criteria for evaluating the two MODM methods and comparing them with each other.

The criterion of mean deviations from the ideal solution (MID criterion): in this criterion mentioned by Coello et al. (Coello, Lamont and Veldhuizen [31]) in 2007, first, an ideal answer to the problem is considered and then the mean deviation of Pareto set of solutions is calculated from the ideal solution. The ideal solution, which is represented by symbol I_{sol}, is a mode in which both solutions are in their optimal state simultaneously $I_{\text{sol}} = (\min(z_1), \min(z_2))$, or in minimization problems, the origin of the coordinates can be considered as the ideal solution ($I_{\text{sol}} = (0,0)$). If $F(A)$ represents the Pareto front of the solution method A, then the MID criterion is calculated as follows: where $\|I_{\text{sol}} - pa\|_2$ represents the Euclidean distance of the solution $pa\ \varepsilon\ F(A)$ from the ideal solution. It is clear that the lower the MID criterion, the better (Behnamian et al. [32]).

$$MI\ D(A) = \frac{\sum_{pa\in F(A)} \|pa - I_{\text{sol}}\|_2}{\|F(A)\|}. \tag{28}$$

Maximum diversity criterion: it measures the length of the space cube diameter used by the end values of the objectives for the set of nondominated solutions, which is calculated in the following equation. The larger the value, the better [8].

$$\text{Diversity} = \sqrt{\left(\max f_{1i} - \min f_{1i}\right)^2 + \left(\max f_{2i} - \min f_{2i}\right)^2}. \tag{29}$$

Uniformity or spacing criterion: it calculates the relative distance of consecutive solutions using the following equation. The smaller the spacing, the better (Schott [33]) (Kaveh and Mahdavi [34]).

$$\text{Spacing} = \left(\frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} \left(d_i - \overline{d}\right)^2\right)^{1/2}, \text{where } \overline{d} = \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} d_i. \tag{30}$$

Criteria of the spread of nondominated solutions (SNS criterion): this indicator can be expressed as an indicator of diversity as follows:

$$\text{SNS} = \sqrt{\frac{\sum_{i=1}^{n} (\text{MID} - C_i)^2}{n-1}}. \tag{31}$$

The higher the SNS value, the better the algorithm. Criterion of rate of achievement to the two objectives simultaneously (RAS): another evaluation criterion proposed in this research is RAS. The RAS equation is shownas follows:

$$\text{RAS} = \frac{\sum_{i=1}^{n} (f_{1i} - F_i/F_i) + (f_{2i} - F_i/F_i)}{n}. \tag{32}$$

In Relation (32), $F_i = \min\{f_{1i}, f_{2i}\}$. The lower the RAS value, the better. It also considers positive objective value in all indicators; therefore, negative objectives must be turned into the positive part of the axis. Each problem is now implemented 10 times for each algorithm, and the graph of

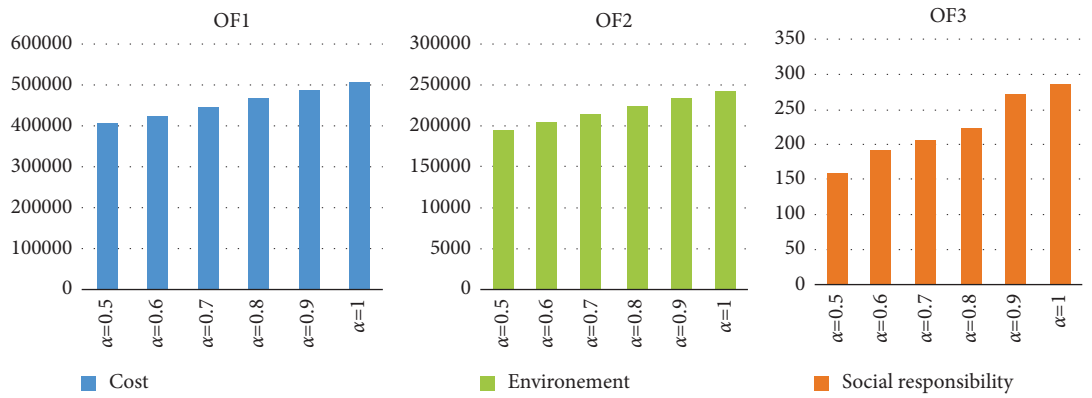


Figure 3: Sensitivity analysis of α.

Table 6: Defined levels for the genetic algorithm parameters to solve problems.

GA parameter/factor	Low level (1)	Intermediate level (2)	High level (3)
Max it	100	150	200
Npop	50	75	100
PC	0.7	0.8	0.9
PM	0.2	0.3	0.4

Table 7: Nature of the research indicators.

Q_i	+ x_1 (diversity)	− x_2 ($MI\ D$)	+ x_3 (SNS)	− x_4 (RAS)
Q_1	r_{11}	r_{12}	r_{13}	r_{14}
Q_2	r_{21}	r_{22}	r_{23}	r_{24}
Q_i	r_{ij}	r_{ij}	r_{ij}	r_{ij}

Table 8: Response values for the NSGA-II algorithm.

Response	1.6592	2.1457	1.2587	1.3624	1.5147	1.1478	1.5214	1.6985	1.4787

Table 9: Result of setting the parameters of NSGA-II algorithm by Taguchi method.

PM	PC	Npop	Max it
0.3	0.8	75	100

Table 10: Defined levels for MOGWO algorithm parameters to solve problems.

GA parameter/factor	Low level (1)	Intermediate level (2)	High level (3)
Max it	100	150	200
Greywolves	50	100	150
A	0.5	0.75	1

Table 11: Response values for the MOGWO algorithm.

Response	1.3201	1.6515	2.1475	1.8711	2.1445	1.9874	2.1142	1.5446	2.1014

Table 12: Result of setting parameters of MOGWO algorithm by Taguchi method.

Max it	Greywolves	A
150	100	0.75

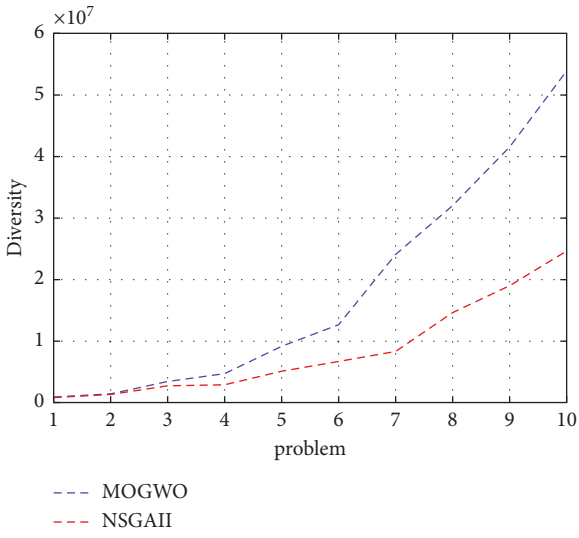


Figure 4: Diversity indicator for the research algorithms.

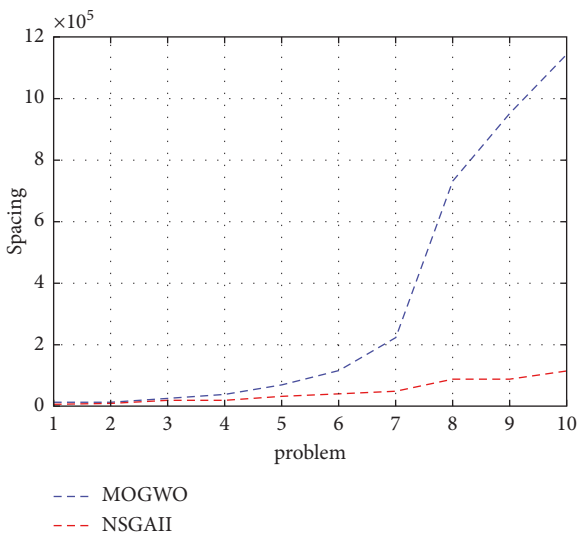


Figure 5: MID indicator for the research algorithms.

each indicator for the two MOGWO and NSGA-II methods is shown below.

The indicator we are examining is the diversity indicator. Figure 4 shows the graph of each algorithm as a comparison diagram of both algorithms according to this indicator. Considering the shape and nature of this indicator, the higher it is, the better. It is clear that the MOGWO algorithm performed better than the NSGA-II algorithm.

For the MID indicator, any algorithm that has a smaller value performs better. According to Figure 5, the MOGWO algorithm is always better than the NSGA-I.

In terms of Spacing indicator, it can be said that the NSGA-II algorithm performed better than the MOGWO algorithm, which is shown in Figure 6.

The higher the SNS indicator, the better. According to Figures 7 and 8, it can be concluded that the MOGWO algorithm is better than the NSGA-II.

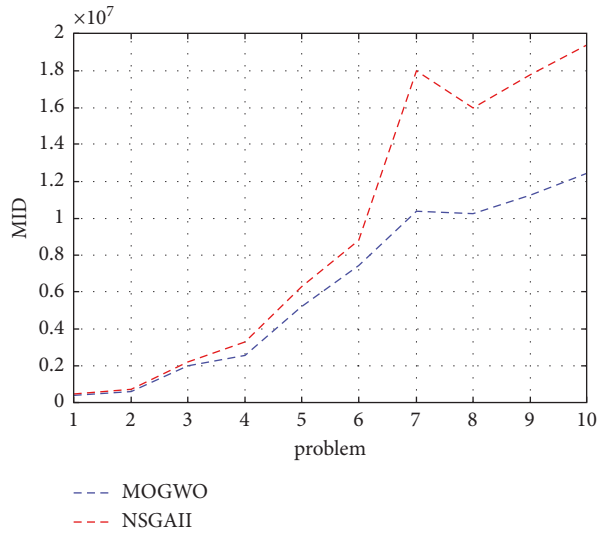


Figure 6: Spacing indicator for the research algorithms.

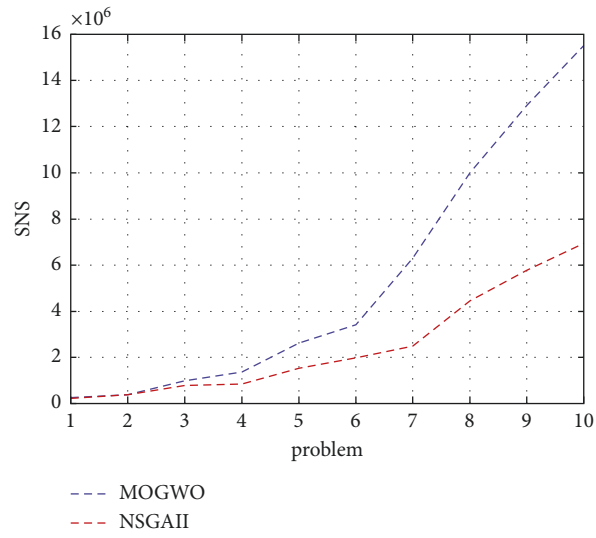


Figure 7: SNS indicator for the research algorithms.

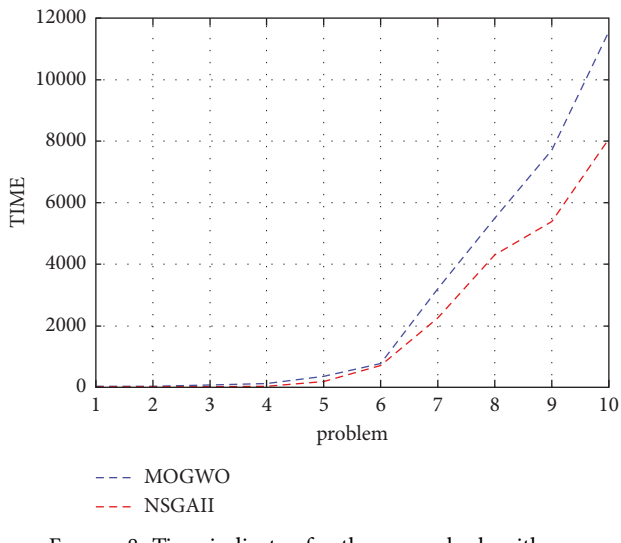


Figure 8: Time indicator for the research algorithms.

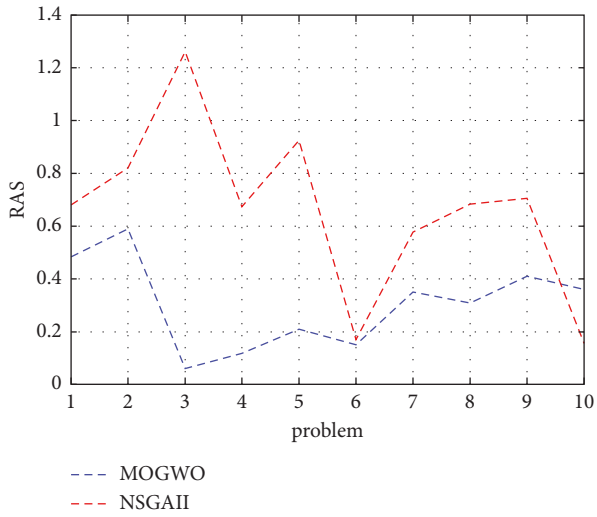


Figure 9: RAS indicator for the research algorithms.

The last indicator studied in this research is the rate of achievement to the two objectives simultaneously (RAS) to solution, which, as mentioned earlier, has a negative nature (the less - the better), i.e. any algorithm that has a lower RAS value is better. According to Figure 9, it is clear that the MOGWO algorithm performed better.

5. Conclusion

In this research, a three-objective mathematical model was stated, the first of which was economic (cost), the second was environmental (air pollution), and the third was social (responsibility). All three objectives were minimization. The model had 14 assumptions and 17 constraints, and the demand for products was fuzzy. The model in the example taken from the real world of manufacturing (manufacturing of safe scaffolding that has all three dimensions of sustainable manufacturing) was solved by GAMZ software through the exact augmented epsilon constraint method (AEC). Due to the fact that the model is NP-hard, meta-heuristic algorithms were used to solve the model. Due to the high complexity of the problem, two algorithms, NSGA-II and MOGWO, were used to solve the proposed model. Finally, 10 problems with different dimensions were solved to evaluate the performance of these two algorithms in MATLAB software and the performance of these two algorithms was determined. The output of the algorithms has produced Pareto fronts close to the Pareto front of the exact method, which is the augmented epsilon constraint (AEC). Sensitivity analysis of α was carried out, and we increased the alpha value step by step. As α increases due to the effect on the pessimistic value of the fuzzy parameters and the nature of the minimization of the objective functions, these functions increase in value. This means that with increasing demand for products and increasing the cost of installation and purchase of machinery and fixed and variable production costs, costs, carbon dioxide emissions, and shortages increase. Finally, we carried out statistical analysis and comparison of two meta-heuristic algorithms. First, four parameters of MaxIt, NPOP, PC, PM for two algorithms were set at optimal levels by adjusting the parameter by Taguchi method, and the results of experimental problems of five indicators were obtained and the output of the mini-tab software for adjusting the parameters of the two algorithms was shown by Taguchi method. The criterion of mean deviations from the ideal solutions (MID), maximum diversity indicator, uniformity indicator, spread of nondominated solutions indicator, and the achievement of two objective functions simultaneously were the five indicators through which we compared the two algorithms. Ten problems were designed to compare algorithms. Then, each problem was implemented ten times for each algorithm, and the average was reported as the final answer for each solution method, which showed the superiority of the multiobjective gray wolf algorithm.

Social, economic, and environmental dimensions each have thousands of criteria. As a result, there is a lot of space for conducting research in this field. The concepts of sustainable production are the newest concepts in manufacturing systems and are not limited to manufacturing industries, and research in this field conducted in recent years confirms this important point, as noted in the review of the research literature, developed industrialized countries have made these concepts binding in all public and private departments and monitor their proper implementation. It is suggested that an institution be established in our country or that there be a department in every organization and manufacturing unit to help institutionalize these concepts in the country while researching in this field and stating the required instructions. Existence of different manufacturing methods can include the dimensions of sustainable production in them, and the benefits of sustainable production can be seen in them. Other meta-heuristic methods can be used to solve the model.

Data Availability

The data used are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. P. Chobar, M. A. Adibi, and A. Kazemi, "Multi-objective Hub-Spoke Network Design of Perishable Tourism Products Using Combination Machine Learning and Meta-Heuristic Algorithms," *Environment, Development and Sustainability*, pp. 1–28, 2022.

[2] A. Heidari, D. M. Imani, and M. Khalilzadeh, "A hub location model in the sustainable supply chain considering customer segmentation," *Journal of Engineering, Design and Technology*, vol. 19, no. 6, pp. 1387–1420, 2021.

[3] P. Ghasemi, H. A. Khalili, A. P. Chobar, S. Safavi, and F. M. Hejri, "A New Multiechelon Mathematical Modeling for Pre-and Postdisaster Blood Supply Chain: Robust Optimization Approach," *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 2976929, 10 pages, 2022.

[4] M. Saidi-Mehrabad and N. Safaei, "A new model of dynamic cell formation by a neural approach," *International Journal of Advanced Manufacturing Technology*, vol. 33, no. 9–10, pp. 1001–1009, 2007.

[5] N. Safaei and R. Tavakkoli-Moghaddam, "An extended fuzzy parametric programming-based approach for designing cellular manufacturing systems under uncertainty and dynamic conditions," *International Journal of Computer Integrated Manufacturing*, vol. 22, pp. 538–548, 2009.

[6] S. Ah Kioon, A. A. Bulgak, and T. Bektas, "Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration," *European Journal of Operational Research*, vol. 192, no. 2, pp. 414–428, 2009.

[7] R. Kia, A. Baboli, N. Javadian, R. Tavakkoli-Moghaddam, M. Kazemi, and J. Khorrami, "Solving a group layout design model of a dynamic cellular manufacturing system with alternative process routings, lot splitting and flexible reconfiguration by simulated annealingflexible reconfiguration by simulated annealing," *Computers & Operations Research*, vol. 39, no. 11, pp. 2642–2658, 2012.

[8] M. Rabbani, F. Jolai, N. Manavizadeh, F. Radmehr, and B. Javadi, "Solving a bi-objective cell formation problem with stochastic production quantities by a two-phase fuzzy linear programming approach," *The International Journal of Advanced Manufacturing Technology*, Springer, vol. 58, pp. 709–722, 2012.

[9] S. S. Ahiska and E. Kurtul, "Modeling and analysis of a product substitution strategy for a stochastic manufacturing/remanufacturing system," *Computers & Industrial Engineering*, vol. 72, pp. 1–11, 2014.

[10] A. Moldavska, "Model-based sustainability assessment – an enabler for transition to sustainable manufacturing," *Procedia CIRP*, vol. 48, pp. 413–418, 2016.

[11] J. Ma, J. D. Harstvedt, D. Dunaway, L. Bian, and R. Jaradat, "An exploratory investigation of Additively Manufactured Product life cycle sustainability assessment," *Journal of Cleaner Production*, vol. 192, pp. 55–70, 2018.

[12] S. Singh, E. U. Olugu, and S. N. Musa, "Development of sustainable manufacturing performance evaluation expert system for small and medium enterprises," *Procedia CIRP*, vol. 40, pp. 608–613, 2016.

[13] Y. Koren, X. Gu, F. Badurdeen, and I. S. Jawahir, "Sustainable living factories for next generation manufacturing," *Procedia Manufacturing*, vol. 21, pp. 26–36, 2018.

[14] M. Mutingi, P. Dube, and C. Mbohwa, "A modular product design approach for sustainable manufacturing in A fuzzy environment," *Procedia Manufacturing*, vol. 8, pp. 471–478, 2017.

[15] M. A. Salido, J. Escamilla, F. Barber, and A. Giret, "Rescheduling in job-shop problems for sustainable manufacturing systems," *Journal of Cleaner Production*, vol. 162, pp. 121–132, 2017.

[16] K. Singh and I. A. Sultan, "Modelling and evaluation of KPIs for the assessment of sustainable manufacturing: an extrusion process case study," *Materials Today Proceedings*, vol. 5, no. 2, pp. 3825–3834, 2018.

[17] O. Fisher, N. Watson, L. Porcu, D. Bacon, M. Rigley, and R. L. Gomes, "Cloud manufacturing as a sustainable process manufacturing route," *Journal of Manufacturing Systems*, vol. 47, pp. 53–68, 2018.

[18] T. Peng, K. Kellens, R. Tang, C. Chen, and G. Chen, "Sustainability of additive manufacturing: an overview on its energy demand and environmental impact," *Additive Manufacturing*, vol. 21, pp. 694–704, 2018.

[19] A. Moldavska and K. Martinsen, "Defining sustainable manufacturing using a concept of attractor as a metaphor," *Procedia CIRP*, vol. 67, pp. 93–97, 2018.

[20] J. Malek and T. N. Desai, "Prioritization of sustainable manufacturing barriers using Best Worst Method," *Journal of Cleaner Production*, vol. 226, pp. 589–600, 2019.

[21] D. Beekaroo, D. S. Callychurn, and D. K. Hurreeram, "Developing a sustainability index for Mauritian manufacturing companies," *Ecological Indicators*, vol. 96, no. 2019, pp. 250–257, 2019.

[22] J. Zheng, X. Zhou, Y. Yu, J. Wu, W. Ling, and H. Ma, "Low carbon, high efficiency and sustainable production of traditional manufacturing methods through process design strategy: improvement process for sand casting defects," *Journal of Cleaner Production*, vol. 253, 2019.

[23] E. Pagone, K. Salonitis, and M. Jolly, "Automatically weighted high-resolution mapping of multi-criteria decision analysis for sustainable manufacturing systems," *Journal of Cleaner Production*, vol. 257, Article ID 120272, 2020.

[24] A. Pourghader Chobar, M. A. Adibi, and A. Kazemi, "A novel multi-objective model for hub location problem considering dynamic demand and environmental issues," *Journal of Industrial Engineering and Management Studies*, vol. 8, no. 1, pp. 1–31, 2021.

[25] M. Rezaei Kallaj, M. Abolghasemian, S. Moradi Pirbalouti, M. Sabk Ara, and A. Pourghader Chobar, "Vehicle routing problem in relief supply under a crisis condition considering Blood types," *Mathematical Problems in Engineering*, vol. 2021, Article ID 7217182, 10 pages, 2021.

[26] A. Pourghader Chobar, M. Sabk Ara, S. Moradi Pirbalouti, M. Khadem, and S. Bahrami, "A multi-objective location-routing problem model for multi-device relief logistics under uncertainty using meta-heuristic algorithm," *Journal of Applied Research on Industrial Engineering*, 2021.

[27] F. Maadanpour Safari, F. Etebari, and A. Pourghader Chobar, "Modelling and optimization of a tri-objective Transportation-Location-Routing Problem considering route reliability: using MOGWO, MOPSO, MOWCA and NSGA-II," *Journal of Optimization in Industrial Engineering*, vol. 14, no. 2, pp. 99–114, 2021.

[28] M. Rahmaty, A. Daneshvar, F. Salahi, M. Ebrahimi, and A. P. Chobar, "Customer churn modeling via the grey wolf optimizer and ensemble neural networks," *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 9390768, 12 pages, 2022.

[29] M. Fazle Baki, B. A. Chaouch, and W. Abdul-Kader, "A heuristic solution procedure for the dynamic lot sizing problem with remanufacturing and product recovery," *Computers & Operations Research*, vol. 43, pp. 225–236, 2014.

[30] H. Bayram and R. Şahin, "A comprehensive mathematical model for dynamic cellular manufacturing system design and Linear programming embedded hybrid solution techniques," *Computers & Industrial Engineering*, vol. 91, pp. 10–29, 2016.

[31] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, "Evolutionary Algorithms for Solving Multi-Objective Problems," Springer, Berlin, Germany, 2007.

[32] J. Behnamian, S. F. Ghomi, and M. Zandieh, "A multi-phase covering pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid

metaheuristic," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11057–11069, 2009.

[33] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," *Doctoral Dissertation, Massachusetts Institute of Technology*, Cambridge, MA, USA, 1995.

[34] A. Kaveh and V. R. Mahdavi, "Multi-objective colliding bodies optimization algorithm for design of trusses," *Journal of Computational Design and Engineering*, vol. 6, no. 1, pp. 49–59, 2019.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9374946, 52 pages
https://doi.org/10.1155/2022/9374946

Research Article

Recall Network: A Simple Brain-Inspired Algorithm for Classification

Zhaoning Tian,[1] Ying Li,[1] Zhenhua Li ,[1] and Site Li[2]

[1]*School of Computer, China University of Geosciences (Wuhan), Wuhan 430074, China*
[2]*Apple Incorporated Company, Cupertino, CA 95014, USA*

Correspondence should be addressed to Zhenhua Li; zhli@cug.edu.cn

Received 15 February 2022; Revised 9 June 2022; Accepted 24 June 2022; Published 13 August 2022

Academic Editor: Diego Oliva

The latest development of neuroscience has deepened the understanding of the information-processing mechanisms in the human brain and inspired a couple of sophisticated computational methods, such as deep learning, memory networks, and hierarchical temporal memory. However, it remains a challenge to explore simpler models due to the high computational cost of the above-mentioned methods. This paper proposes recall network (RN), an intuitive and simple model, that initializes itself by constructing the network path derived from the correlation of features in the training dataset and then makes classification decisions by recalling the paths that are relevant to the features in the test set. The algorithm has been applied to 263 datasets available from UCI Machine Learning Repository, and the classification results of repeated 10-fold cross-validation experiments on Weka demonstrate its competitive performance with prestigious classification algorithms, such as ANN, J48, and KNN.

1. Introduction

Brain-inspired algorithms, like artificial neural networks, have shown great success in solving numerous problems in multiple fields for many years. Recently, the development of brain science and neuroscience has deepened the understanding of brain information-processing mechanisms and therefore inspires new generation computational models. For example, deep learning surpasses artificial neural networks (ANNs) in terms of both complexity and capability [1–3], and later in 2015, memory networks (MemNN) came with a long-term memory component [4, 5]; on the other hand, hierarchical temporal memory (HTM) builds a tree-shaped hierarchy of levels where the higher level gets input from results from the next lower level [6, 7]. Bin Hu etc. suggested that it is a feasible way to reconstruct cortical networks with dynamic activities instead of using only artificial computing networks [8].

Those algorithms have a trend of being more and more sophisticated. However, designing simpler models is still worth exploring due to the high computational cost of the above-mentioned methods. This work proposes a straightforward model called recall network (RN), which is a memory network that stores, marks, and then retrieves the previous paths. A RN consists of nodes and edges (path), where a node represents an attribute (feature) value, while an edge connects between two nodes and denotes the number of samples that has both attribute values. For each sample, if we connect its attribute values consecutively, we can form an end-to-end route. During the prediction phase, the network determines its type by plural edges along the route. For example, if a route consists of three edges with 2 YES and 1 NO, then the result will be YES.

This paper investigates the use and application of recall network in the realm of classification field. The objective of the study is to unveil the capabilities of this newly developed method on benchmark problems of classification. Compared with other classic approaches, the performance of the proposed algorithm is experimented and evaluated on Weka [9], a prestigious machine learning platform with the function of the fair comparison of algorithm performance. The results obtained verify the promising performance of the proposed algorithm.

The remaining sections of the paper are organized as follows. The second section introduces the proposed algorithm with particular emphasis on its application in the classification field. In the third section, the concept of RN is compared with other similar algorithms. Classification examples proving the accuracy of the propose algorithm are covered in the fourth section. The fifth section discusses the difference between RN and some powerful classification methods. A brief conclusion of the study is given in the last section.

2. Structure and Algorithms

2.1. Structure of a Recall Network. A recall network is a group of connected layers, where every layer consists of similar nodes that represent the value for a certain attribute (see Figure 1). Also, each connection or edge, expressed as a set, stores relations between connected nodes of previous samples, for the purpose of determining the output value of the edge of the pair nodes by their plurality.

RN can be defined as an undirected graph $G = (N, E)$, consisting of the set N of nodes where nodes representing the same attribute are lined in one layer, and the set E of edges, where each edge connects two nodes from adjacent layers.

2.2. Mechanism of Recall Network. Each training sample, called an instance in Weka, is represented by a route from a node in the first layer to the last one. Routes are piled up in the training process, and during the prediction phase, each predicted sample, also a route, is voted by each edge whose value was accumulated in the training stage.

The above mechanism can be expressed as

$$\begin{aligned} f &= \text{Mode}\left(e_1, e_2, e_3, \ldots, e_n\right), \\ e_i &= \text{Mode}\left(e_{i_1}, e_{i_2}, e_{i_3}, \ldots, e_{i_j}\right), \end{aligned} \tag{1}$$

where $e_1, e_2, e_3, \ldots, e_n$ are edges of the instance route from the 1st layer to the last one (n) and the Mode function is to get the mode of the edge set. In the same way, each edge ei gets the mode of plural connections of their adjacent nodes e_{i1}, e_{i2}, e_{i3}, $\ldots$, e_{ij}.

The algorithm is simple to understand and cheap to implement, requiring only three easy steps: building a RN from instances (see Algorithm 1), training it by known data (see Algorithm 2), and using it to classify unknown samples (see Algorithm 3).

2.3. A Classification Example. To illustrate the proposed algorithm, an example is shown in Table 1. In this example, our alrogithm aims to predict whether a given weather condition is suitable for playing tennis.

Figure 2 shows how a RN is created based on Table 1. Here, the RN consists of 4 layers (attributes) with Outlook, Temperature, Humidity, and Wind. Each node represents a possible value for a weather attribute, and nodes that shared the same attribute are arranged in a column. Each edge connects two nodes from adjacent columns. If a given day (a row in Table 1) is suitable for playing tennis (PlayTennis = YES), then we add one blue edge between each adjacent pair of nodes, in which the node should correctly represent the value of a weather attribute on that day. If a given day is not suitable for playing tennis (PlayTennis = NO), we add red edges with similar criteria.

To predict a new sample with following weather condition: Rain, Mild, High, Strong, first, we idenfity the set of 4 nodes that represent that weather condition in Figure 2. Then, we find all the edges (Rain-Mild, Mild-High, High-Strong) that connects each pair of nodes within that set. Finally, we group those edges by color and count them in Figure 2. In this example, we got 3 red edges and 5 blue edges and therefore, this sample is likely to be classified as PlayTennis = YES (5 blues win 3 reds). However, if we got the same number of red and blue edges, the system will choose the color of the first edge in the edge set.

2.4. Appropriated Problems. Apparently, the proposed approach can naturally be used for discrete values and categorical values, and datasets with missing or error values are tolerated.

For continuous values, it divides the data range into a couple of intervals, and each interval is regarded as a discrete value. As for the data missing case, the missing node will be skipped.

3. Conceptual Comparison of RN with Similar Algorithms

Though RN is a novel algorithm, its structure is still similar to artificial neural network (ANN) and the memory method like pheromone in ant colony optimization (ACO); after all, it is a simple, intuitive, and high readability brain-inspired algorithm.

3.1. Compared with the ANN. The RN shares the same biological motivation and similar network structure with ANN; however, there are three main differences.

(1) The meaning of nodes is totally different. A node in ANN represents the summation of inputs while it works only as an identifier in RN.

(2) Structure is more flexible in ANN but not in RN, since the number of RN layers is decided strictly by the number of attributes, and the number of nodes should equal the number of intervals of the attribute.

(3) Though ANN is an effective algorithm in many fields, it is a "black box," while RN is more explainable.

3.2. Compared with the MemNN. MemNN, strictly speaking, is more like a system consisting of a set of memory and inference modules than an independent algorithm. The idea that retrieves the most relevant memory is similar to RN; however, RN is much simpler: it does not need to compress and transform the input or use complex functions to score memories. On the contrary, RN just stores the input as it is and then returns the vote based on the majority marks of the same kind.

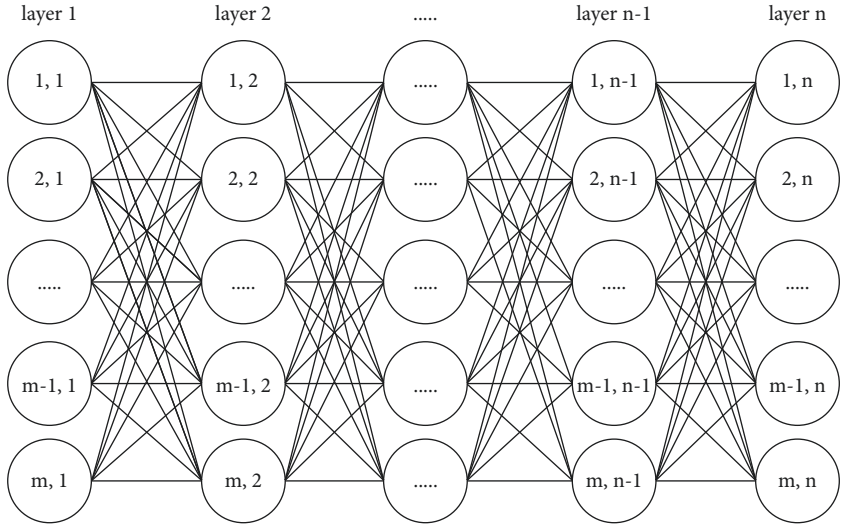


Figure 1: The diagram of a recall network.

Input: number of attributes (N), number of categories for each discrete attribute, scope of value for each continuous attribute
Output: a network with N layers where each layer has different nodes corresponding to the intervals of an attribute
Initialization of the Recall Network;
for $i \longleftarrow 1$ **to** *the Number of attribute*−1 **do**
  for $m \longleftarrow 1$ **to** *the intervals of the* i_{th} *attribute* **do**
    for $n \longleftarrow 1$ **to** *the intervals of the* $(i+1)_{th}$ *attribute* **do**
      Create two nodes: node (i,m) and node (i+1,n);
      Create an edge between them;
      Initializing classID vector of the edge;
    end
  end
end

Algorithm 1: Building a recall network from instances.

input: training data
output: the network weighted by all training examples
Training the Recall Network;
for $i \longleftarrow 1$ **to** *the number of examples* **do**
  classIndex = the class index of the i_{th} example
  for $j \longleftarrow 1$ **to** *the Number of attribute*−1 **do**;
    $RowNo_j$ = the row number of the value in j_{th} layer;
    $RowNo_{j+1}$ = the row number the value in $(j+1)_{th}$ layer;
    e = the edge between the *Node* (j, $RowNo_j$) and *Node* (j, $RowNo_{j+1}$);
    $e.ClassVector[ClassIndex] + 1$
  end
end

Algorithm 2: Teaching the recall network with the training dataset.

3.3. Compared with the HTM. Like HTM, RN simulates the mechanism of the cerebral cortex and memorizes patterns for solving problems, but RN has only one "level" and therefore spares the transformation process from lower levels to higher levels; moreover, RN uses a very primitive method—voting—to identify the patterns.

3.4. Compared with the ACO. RN is similar to ACO in the idea of "pheromone." As long as a sample passes through a network route, each segment (edge) in the route will get a permanent mark, and such a mark will never decay along with the time, unlike ACO. We argue that RN more meets biological characteristics of the brain: memories learned by

input: test data
output: the class of the data
Determining the category by votes from all edges;
for $i \longleftarrow 1$ **to** *the number of attributes*-1 **do**
    $RowNo_i$ = the row number of the value in i_{th} layer;
    $RowNo_{i+1}$ = the row number the value in $(i+1)_{th}$ layer;
    e = the edge between the *Node* $(i, RowNo_i)$ and *Node* $(i, RowNo_{i+1})$;
    $ResultVector += e.ClassVector$
end
return the mode in ResultVector

Algorithm 3: Predicting the class by a trained recall network.

Table 1: A classic example of playing tennis [10].

		Weather			
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

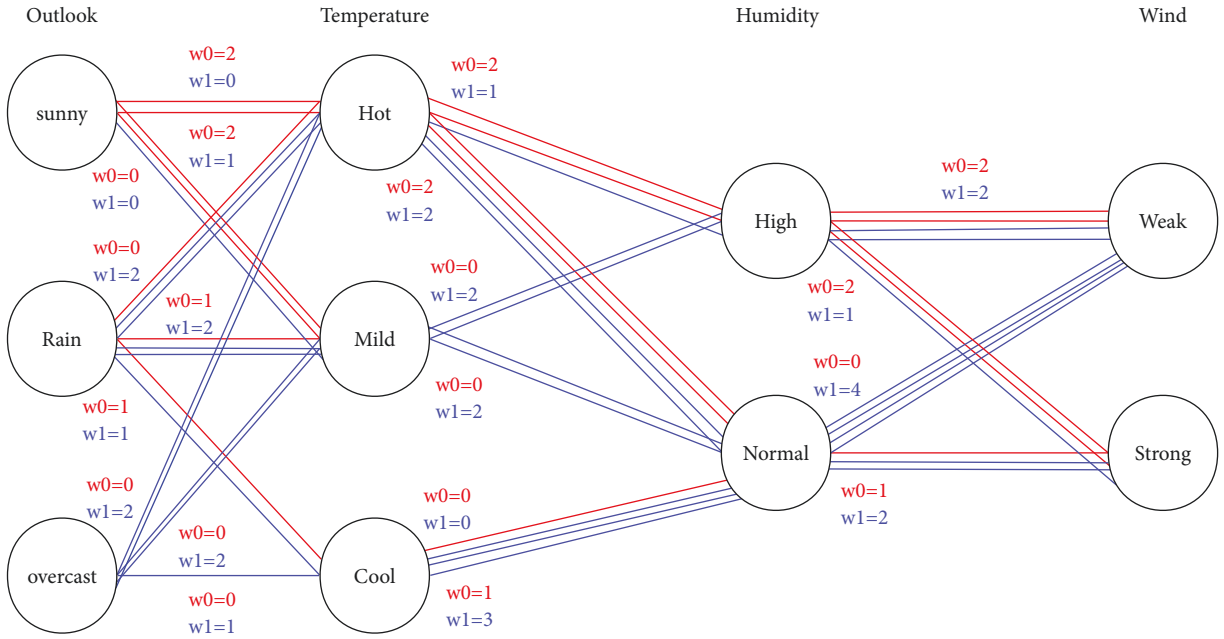
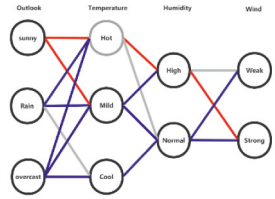
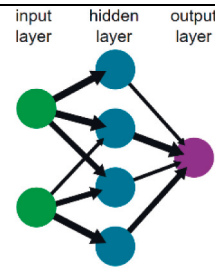
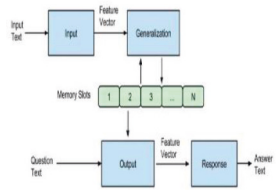
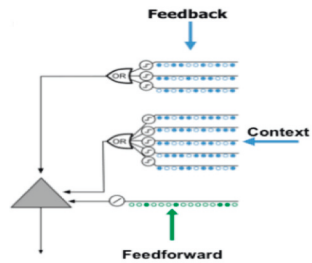
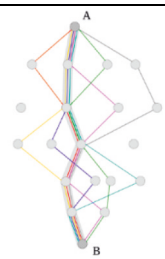


Figure 2: The RN graph corresponding to Table 1.

Table 2: Comparison with similar algorithms in terms of mechanism, complexity, and structure (pictures of NN, MemNN, HTM, and ACO are copied from https://www.wikipedia.com; the time complexity and space complexity of ANN and MemNN are collected from [11–17]).

Algorithm	Mechanism	Time complexity	Space complexity	Structure
RN	Sum up the votes in an end-to-end route	$O(n * d * m)$	$O(n * d)$	
ANN	A linear combination of neurons and weights	$O(\sum_{l=1}^{D} M_l^2 * K_l^2 * C_{l-1} * C_l)$	$O(\sum_{l=1}^{D} K_l^2 * C_{l-1} * C_l + \sum_{l=1}^{D} M^2 * C_l)$	
MemNN	Search the most relevant memory slot step by step until the answer is found	$O(\sum_{l=1}^{D} n * d^2)$	$O(\sum_{l=1}^{D} n + M)$	
HTM	The lower level outputs the generated pattern to higher levels hierarchically	\	\	
ACO	Good paths are encouraged to repeat while random exploration is still maintained	\	\	

brain will hardly disappear completely, and they will be recalled when a similar event triggers even through a long period.

3.5. Comparison of the Mechanism, Computing Complexity, and Structure. To distinguish the RN further, we analyze RN in terms of mechanism, time complexity, space complexity, and structure (Table 2).

In a summary, RN is a distinctive brain-inspired algorithm. Although its network structure is similar to ANN, and its memory mechanisms is similar to MemNN, HTM, and ACO, RN is the simplest of them all.

4. Experiments and Results

4.1. Testing Tool. To evaluate the performance of the proposed algorithm, we chose Weka, a prestigious machine learning platform that gives a fair comparison among different algorithms. No doubt, Weka is one of the most popular tools to research machine learning algorithms. A number of scientists use this software to study classification problems, for example, Arora and Suman tested J48 and MLP on 5 UCI datasets [18]; Kiranmai and Laxmi classified power quality problems using Weka and studied the effect of attributes on classification accuracy [19]; Mhetre and Nagar adopted Weka to compare four classification algorithms (Naive Bayes, J48, ZeroR, and

random tree) on education datasets [20]; Farhat et al. compared the performance of SVM, KNN, NB, logistic regression, decision tree, and random forest on intrusion detection data [21]; Villavicencio et al. studied J48 decision tree, random forest, SVM, KNN, and NB on the COVID-19 Symptoms and Presence dataset from Kaggle [22].

It can be seen from the above research that SVM, KNN, J48, NB, and OneR are competitive methods in the classification field. However, the above research works focus on limited field or limited datasets. In this paper, we used all available data to study the performance comparison.

4.2. Testing Data. The test problem includes 263 UCI datasets from UCI Machine Learning Repository. Although the repository offers 466 datasets in total, 121 are too big for Weka, 20 are missing, 53 are for natural language processing, and 9 are pictures. Somehow, the rest still represent a wide range of domains and data characteristics, whose detailed description is listed in Table 3. In a word, all available data in UCI Repository are used for testing.

4.3. Method. The RN is published on https://github.com/tianzhaoning/RRN/releases/download/RecallNetwork/RN.zip, and we then retrieved them back to Weka on a local machine to compare with others. In all experiments, 10 runs of 10-fold cross validation are used to average the results.

To illustrate the performance of the proposed method, we chose some competitive classification algorithms, such as Naive Bayes (NB), support vector machine (SVM), decision tree (J48), K-nearest neighbors (KNN), artificial neural network (ANN), and OneR (since it is also a very simple decision tree algorithm), on a bulk of benchmark data.

All algorithms use default parameters supplied by Weka, for the purpose of fair play. However, we applied two more parameters on SVM and one more on ANN besides the default setting to explore the effect under different configurations.

As for SVM, we took the default as the SVM1, changing the following parameters: degree, gamma, cachesize, cost, E-eps, and the P value (the value of the loss function in e-SVR). As for ANN, we took the default as ANN1, we change the learning rate and the momentum. The specific values of super-parameters are shown in Table 4.

In the process of statistical analysis, this paper adopts the Friedman test, which is a non-parametric test method using rank to realize whether there are significant differences in multiple population distributions, and ANOVA to test the remarkable mean difference among multiple groups. If both results are yes, we then use the Tukey HSD test to make pairwise comparisons.

4.4. Performance Metrics. Totally, we evaluate the classification performance with the following indicators [23, 24]:

(1) Accuracy (ACC): number of correctly predicted items divided by the total of the item to predict. It is the most important metric to measure classification performance. In this work, we use mean ACC: the average of each accuracy per dataset, which represents the average performance over all tasks.

(2) Kappa statistic (k): a measure of how much better the classification results are compared to classification labels assigned by random chance. Generally, in terms of degree of agreement, $k \leq 0$ is interpreted as indicating no agreement, 0.01–0.20 as none to slight, 0.21–0.40 as fair, 0.41–0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 as almost perfect agreement.

(3) Root mean squared error (RMSE): the standard deviation of the residuals (prediction errors). This is another indicator of model accuracy.

(4) Mean absolute error (MAE): the average difference between the prediction result and the true value. MAE can avoid the problem of error cancellation, so it can reflect the actual prediction error.

(5) Relative absolute error (RAE): sum of the absolute value of the forecast errors divided by the sum of the absolute values of the mean errors. The RAE considers each error equally important. The better the model is, the closer its RAE is to 0.

(6) Root relative squared error (RRSE): square root of the relative squared error (RSE) where RSE means the sum of squared errors of a predictive model normalized by the sum of squared mean errors. It tells how well a model performs relative to the average of the true values.

(7) Running time: the executed time of an algorithm on data. It is an intuitive measurement of the time complexity in the real environment. In this paper, we use total time instead of running time per dataset, since Weka does not provide the detailed time on each experiment.

Generally, 10 runs of 10-fold cross validation must be used to average the above results [25] with the exception of run time, which uses the total time.

4.5. Results. Table 5 and Figure 3 show the average performance of each algorithm on all datasets. The detailed results are listed in Tables 6–11 which present the accuracy (ACC), kappa statistic (k), root mean squared error (RMSE), mean absolute error (MAE), relative absolute error (RAE), root relative squared error (RRSE), and their standard deviation of each model on each dataset, respectively.

Then, we used Friedman and ANOVA tests to compare the above 6 performance metrics of all models among 263 datasets, and the tests proved that all algorithms differed in both the distribution and the mean value at 95% confidence level (see Table 5). Then, the Tukey HSD test is employed to determine which pair has significant differences, and the detailed result is presented in Tables 12–17, where the result underlined means the corresponding pair is remarkably different.

Some findings of RN are summarized briefly as follows:

(1) Figure 3 shows that the performance of RN is not significantly different from other prestigious classification algorithms.

Table 3: Description of datasets.

No	Dataset	Examples	Attributes	Classes	Missingvalue	Numricvalue
1	Abalone	4177	9	3	N	Y
2	Absenteeism-at-work	740	20	2	N	Y
3	Acute-inflammation	120	7	2	N	Y
4	Acute-nephritis	120	7	2	N	Y
5	Adult	1628	15	2	N	Y
6	Aggregation	788	3	7	N	Y
7	Algerianforest	243	14	2	N	Y
8	Annealing	798	32	5	N	Y
9	Arrhythmia	452	263	13	N	Y
10	Au1 1000	1000	21	2	N	Y
11	Au4-2500	1000	101	3	N	Y
12	Au6-1000	1000	41	8	N	Y
13	Au6-250-drift-au6-cd1-500	750	41	8	N	Y
14	Au6-cd1-400	400	41	8	N	Y
15	Au7-300-drift-au7-cpd1-800	1100	13	5	N	Y
16	Au7-700	700	13	5	N	Y
17	Au7-cpd1-500	500	13	5	N	Y
18	Audiology-std	171	60	18	N	Y
19	Audit-risk	776	25	2	Y	Y
20	Autism-adolescent-data	104	21	2	Y	N
21	Autism-adult-data	704	21	2	Y	N
22	Autism-child-data	292	21	2	Y	N
23	Autos	205	26	7	Y	Y
24	Avila	4172	11	12	N	Y
25	Balance-scale	625	5	3	N	Y
26	Balloons	16	5	2	N	Y
27	Bank	4521	17	2	N	Y
28	Blood	748	5	2	N	Y
29	Breast-cancer-wisc-diag	569	31	2	N	Y
30	Breast-cancer-wisc-prog	198	34	2	N	Y
31	Breast-cancer-wisc	699	10	2	N	Y
32	Breast-cancer	286	10	2	N	Y
33	Breast-tissue	106	10	6	N	Y
34	Bupa	345	7	2	N	Y
35	Caesarean	80	6	2	N	N
36	Car	1728	7	4	N	Y
37	Cardiotocography-10classes	2126	22	10	N	Y
38	Cardiotocography-3classes	2126	22	3	N	Y
39	Cervical-cancer	72	20	2	N	Y
40	Chemicalcomposionofceramic	88	18	2	N	Y
41	Chess-krvk	2805	7	18	N	Y
42	Chess-krvkp	3196	37	2	N	Y
43	Congressional-voting	435	17	2	N	Y
44	Conn-bench-sonar-mines-rocks	208	61	2	N	Y
45	Conn-bench-vowel-deterding	528	12	11	N	Y
46	Connect-4	3377	43	2	N	Y
47	Connectionist	207	61	2	N	Y
48	Contrac	1473	10	3	N	Y
49	COVID-19	14	8	3	N	N
50	Credit-approval	690	16	2	N	Y
51	Crowdsource	2109	29	6	N	Y
52	Crx	690	16	2	Y	N
53	Cryother	90	7	2	N	Y
54	Cylinder-bands	512	36	2	N	Y
55	Dbworld-bodies	64	4703	2	N	N
56	Dbworld-bodies-stemmed	64	3722	2	N	N
57	Dbworld-subjects	64	243	2	N	N
58	Dbworld-subjects-stemmed	64	230	2	N	N
59	Dermatology	366	35	6	N	Y
60	Diabetes	768	9	2	N	Y

Table 3: Continued.

No	Dataset	Examples	Attributes	Classes	Missingvalue	Numricvalue
61	Diabetic	1151	20	2	N	Y
62	Divorce	170	55	2	N	N
63	Dota2train	926	116	2	N	N
64	Dow-jones-index	749	15	2	Y	Y
65	Dry-bean-dataset	3402	17	7	N	Y
66	Early-stage-diabetes-data-upload	520	17	2	N	N
67	Echocardiogram	131	11	2	N	Y
68	Ecoli	336	8	8	N	Y
69	Eegeyesate	14980	15	2	N	Y
70	Electrical	312	14	2	N	Y
71	Energy-y1	768	9	3	N	Y
72	Energy-y2	768	9	3	N	Y
73	Extentionofz-alizadehsani	303	59	2	N	Y
74	Fertility	100	10	2	N	Y
75	First-order	1529	57	2	N	Y
76	Flags	194	29	8	N	Y
77	Foresttypes	198	28	4	N	Y
78	Garments-worker-productivity	1197	14	5	Y	Y
79	Gender-name-dataset	4602	3	2	N	Y
80	Gesture-a1-raw	1747	20	5	N	Y
81	Gesture-a1-va3	1743	33	5	N	Y
82	Gesture-a2-raw	1264	20	5	N	Y
83	Gesture-a2-va3	1260	33	5	N	Y
84	Gesture-a3-raw	1834	20	5	N	Y
85	Gesture-a3-va3	1830	33	5	N	Y
86	Gesture-b1-raw	1073	20	6	N	Y
87	Gesture-b1-va3	1069	33	5	N	Y
88	Gesture-b3-raw	1424	20	5	N	Y
89	Gesture-b3-va3	1420	33	5	N	Y
90	Gesture-c1-raw	1111	20	5	N	Y
91	Gesture-c1-va3	1107	33	5	N	Y
92	Gesture-c3-raw	1448	20	5	N	Y
93	Gesture-c3-va3	1444	33	5	N	Y
94	Glass	214	10	6	N	Y
95	Go-track-tracks	163	9	2	N	Y
96	Haberman-survival	306	4	2	N	Y
97	Hayes-Roth	132	4	3	N	Y
98	Hcc-data	165	50	2	Y	N
99	Hcvdat	615	13	5	N	Y
100	Heart-Cleveland	303	14	5	N	Y
101	Heart-Hungarian	294	13	2	N	Y
102	Heart-Switzerland	123	13	5	N	Y
103	Heart-va	200	13	5	N	Y
104	Heart-failure-clinical-records-dataset	299	13	2	N	Y
105	Hepatitis	155	20	2	N	Y
106	Hill-valley	606	101	2	N	Y
107	Hiv1625data	1625	9	2	N	N
108	Hiv746data	746	9	2	N	N
109	Horse-colic	300	26	2	N	Y
110	Htru	17898	9	2	N	Y
111	Hypothyroid	3772	30	4	Y	N
112	Ibeacon-rssi-labeled	1420	14	105	N	Y
113	Ilpd-Indian-liver	583	10	2	N	Y
114	Image-segmentation	210	19	7	N	Y
115	Immunotherapy	90	8	2	N	Y
116	Impensdata	947	9	2	N	N
117	In-vehicle-coupon-recommendation	12684	24	2	Y	N
118	Indian	583	11	2	Y	Y
119	Ionosphere	351	34	2	N	Y
120	Iris	150	5	3	N	Y

Table 3: Continued.

No	Dataset	Examples	Attributes	Classes	Missingvalue	Numricvalue
121	Jain	372	3	2	N	Y
122	Jsbach-chorals-harmony	5665	15	2	N	N
123	Knowledge	172	6	4	N	Y
124	Lasvegastripadvisorreviews	504	19	7	N	N
125	Leaf	340	16	30	N	Y
126	Led-display	1000	8	10	N	Y
127	Lenses	24	5	3	N	Y
128	Letter	20000	17	26	N	Y
129	Libras	360	91	15	N	Y
130	Low-res-spect	531	101	9	N	Y
131	Lung-cancer	32	57	3	N	Y
132	Lymphography	148	19	4	N	Y
133	Magic	1902	11	2	N	Y
134	Mammographic	961	6	2	N	Y
135	Miniboone	1300	51	2	N	Y
136	Molec-biol-promoter	106	58	2	N	Y
137	Molec-biol-splice	3190	61	3	N	Y
138	Monks-1	124	7	2	N	Y
139	Monks-2	169	7	2	N	Y
140	Monks-3	122	7	2	N	Y
141	Mushroom	8124	22	2	N	Y
142	Musk-1	476	167	2	N	Y
143	Musk-2	3299	167	2	N	Y
144	Newdiagnosis	120	7	2	N	N
145	Nursery	12960	9	5	N	Y
146	Obesitydataset-raw-and-data-sinthetic	2111	17	7	N	N
147	Obs-network-dataset-2-aug27	1075	22	4	Y	Y
148	Occupancy-data	2665	7	2	N	Y
149	Occupancy-data2	9752	7	2	N	Y
150	Occupancy-data3	8143	7	2	N	Y
151	Old	2456	31	2	N	N
152	Online-shoppers-intention	12330	18	2	N	Y
153	Oocytes-merluccius-nucleus-4d	1022	42	2	N	Y
154	Oocytes-merluccius-states-2f	1022	26	3	N	Y
155	Oocytes-trisopterus-nucleus-2f	912	26	2	N	Y
156	Oocytes-trisopterus-states-5b	912	33	3	N	Y
157	Optdigits	5620	65	10	N	Y
158	Optical	3823	63	10	N	Y
159	Ozone	2536	73	2	N	Y
160	Page-blocks	5473	11	5	N	Y
161	Parkingbirmingham	3571	3	30	N	Y
162	Parkinsons	195	23	2	N	Y
163	Pasture	36	23	3	N	Y
164	Pbc	418	19	2	Y	Y
165	Pen	10992	17	10	N	Y
166	Pendigits	7494	17	10	N	Y
167	Pharynx	195	12	2	Y	N
168	Phishingwebsites	11055	31	2	N	N
169	Pima	768	9	2	N	Y
170	Pittsburg-bridges-rel-l	103	8	3	N	Y
171	Pittsburg-bridges-span	92	8	3	N	Y
172	Pittsburg-bridges-t-or-d	102	8	2	N	Y
173	Pittsburg-bridges-type	105	8	6	N	Y
174	Pittsburg-bridgesmaterial	106	8	3	N	Y
175	Planning	182	13	2	N	Y
176	Plant-margin	1600	65	100	N	Y
177	Plant-shape	1600	65	100	N	Y
178	Plant-texture	1599	65	100	N	Y
179	Poker-hand-training-true	25010	11	10	N	Y
180	Post-operative	90	9	3	N	Y

Table 3: Continued.

No	Dataset	Examples	Attributes	Classes	Missingvalue	Numricvalue
181	Primary-tumor	330	18	15	N	Y
182	Qsarbioconcentration	779	12	2	N	Y
183	Qsarbiodegradation	1055	42	2	N	Y
184	Qualitative-bankruptcy	250	7	2	N	N
185	Ringnorm	7400	21	2	N	Y
186	Risk-factors-cervical-cancer	858	36	2	Y	Y
187	Robotnavigation	5456	25	4	N	Y
188	Sapfile	131	22	3	N	N
189	Sat	6435	37	6	N	Y
190	Satellite	6435	37	6	N	Y
191	Scadi	70	206	7	N	N
192	Schillingdata	3272	9	2	N	N
193	Seeds	210	8	3	N	Y
194	Segment	2310	20	7	N	Y
195	Seismic-bumps	2584	19	2	N	Y
196	Semeion	1593	257	10	N	Y
197	Setapprocesst1	64	85	2	N	Y
198	Setapprocesst10	74	85	2	N	Y
199	Setapprocesst11	74	85	2	N	Y
200	Setapprocesst2	74	85	2	N	Y
201	Setapprocesst3	74	85	2	N	Y
202	Setapprocesst4	63	85	2	N	Y
203	Setapprocesst5	74	85	2	N	Y
204	Setapprocesst6	74	85	2	N	Y
205	Setapprocesst7	74	85	2	N	Y
206	Setapprocesst8	74	85	2	N	Y
207	Setapprocesst9	74	85	2	N	Y
208	Shillbiddingdataset	6321	9	2	N	Y
209	Shuttle-landing-control	15	7	2	N	N
210	Somervillehappinesssurvey2015	143	7	2	N	N
211	Sonar	208	61	2	N	Y
212	Soybean	307	36	18	N	Y
213	Spambase	4601	58	2	N	Y
214	Speaker-accent	329	13	6	N	Y
215	Spect	79	23	2	N	Y
216	Spectf	80	45	2	N	Y
217	Statlog-Australian-credit	690	15	2	N	Y
218	Statlog-German-credit	1000	25	2	N	Y
219	Statlog-heart	270	14	2	N	Y
220	Statlog-image	2310	19	7	N	Y
221	Statlog-landsat	4435	37	6	N	Y
222	Statlog-shuttle	43500	10	7	N	Y
223	Statlog-vehicle	846	19	4	N	Y
224	Steel-plates	1941	28	7	N	Y
225	Synthetic-control	600	61	6	N	Y
226	Teaching	151	6	3	N	Y
227	Thoraricsurgery	470	17	2	N	N
228	Thyroid	215	6	3	N	Y
229	Thyroid-train	3772	22	3	N	Y
230	Tic-tac-toe	958	10	2	N	Y
231	Titanic	2201	4	2	N	Y
232	Trains	10	30	2	N	Y
233	Transfusion	748	5	2	N	Y
234	Trial	776	17	2	Y	Y
235	Turkiye-student-evaluation	5820	34	3	N	Y
236	Unbalanced	856	33	2	N	Y
237	Urbanlandcover	168	148	9	N	Y
238	Userknowledgemodeling	252	6	4	N	Y
239	Vehicle	846	19	4	N	Y
240	Vertebral-column-2classes	310	7	2	N	Y

Table 3: Continued.

No	Dataset	Examples	Attributes	Classes	Missingvalue	Numricvalue
241	Vertebral-column-3classes	310	7	3	N	Y
242	Veteran	137	8	2	N	Y
243	Vowel	990	14	11	N	Y
244	Wall-following	5456	25	4	N	Y
245	Waveform-noise	5000	41	3	N	Y
246	Waveform	5000	22	3	N	Y
247	Wbc	683	10	2	N	N
248	Wdbc	569	31	2	N	Y
249	Weathernominal	14	5	2	N	N
250	Weathernumeric	14	5	2	N	N
251	Website-phishingdata	1353	10	3	N	N
252	Wholesalecustomersdata	440	8	2	N	Y
253	Wifi-localization	2000	8	4	N	Y
254	Wilt	4839	6	2	N	Y
255	Wine-quality-red	1599	12	6	N	Y
256	Wine-quality-white	4898	12	7	N	Y
257	Wine	178	14	3	N	Y
258	Yamilnaduelectricty	9156	4	20	N	Y
259	Yeast	1484	9	10	N	Y
260	Youtobe-kabita-preprocessing	4900	8	7	Y	Y
261	Youtobe-nisha-preprocessing	4900	8	7	N	Y
262	Z-alizadehsani	303	56	2	N	Y
263	Zoo	101	17	7	N	Y

Table 4: The parameters of SVM and ANN.

	Degree	Gamma	Cache	Cost	E-eps	P value
SVM1	3	0.5	40	1	0.001	0.1
SVM2	4	0.6	50	2	0.002	0.2
SVM3	5	0.7	60	3	0.001	0.3
		Learning rate			Momentum	
ANN1		0.3			0.2	
ANN2		0.4			0.3	

Table 5: Performance of all models among 263 datasets.

		Average accuracy (%)	Kappa statistic	Root mean squared error	Mean absolute error	Relative absolute error (%)	Root relative squared error (%)	Total running time (i7, 16G)
RN		73.57	0.42	0.33	0.27	79.08	83.25	3,780s
NB		72.05	0.50	0.34	0.19	60.74	85.51	60s
J48		78.53	0.59	0.28	0.16	44.76	70.59	180s
KNN		76.87	0.56	0.32	0.16	45.24	80.47	780s
SVM1		71.96	0.42	0.36	0.17	49.04	90.03	8,100s
SVM2		72.66	0.44	0.36	0.17	47.95	88.26	8,100s
SVM3		73.01	0.45	0.35	0.16	47.48	87.19	8,100s
ANN1		77.95	0.59	0.28	0.15	43.61	69.93	34.080s
ANN2		77.84	0.59	0.28	0.15	43.27	70.66	34.080s
OneR		67.16	0.40	0.39	0.19	56.74	98.86	60s
Friedman ranked test	Q (95% threshold: 21.67)	218.45	320.03	467.68	555.13	575.90	460.46	—
ANOVA	F (95% threshold: 1.88)	8.20	15.05	13.13	16.48	29.35	18.82	—

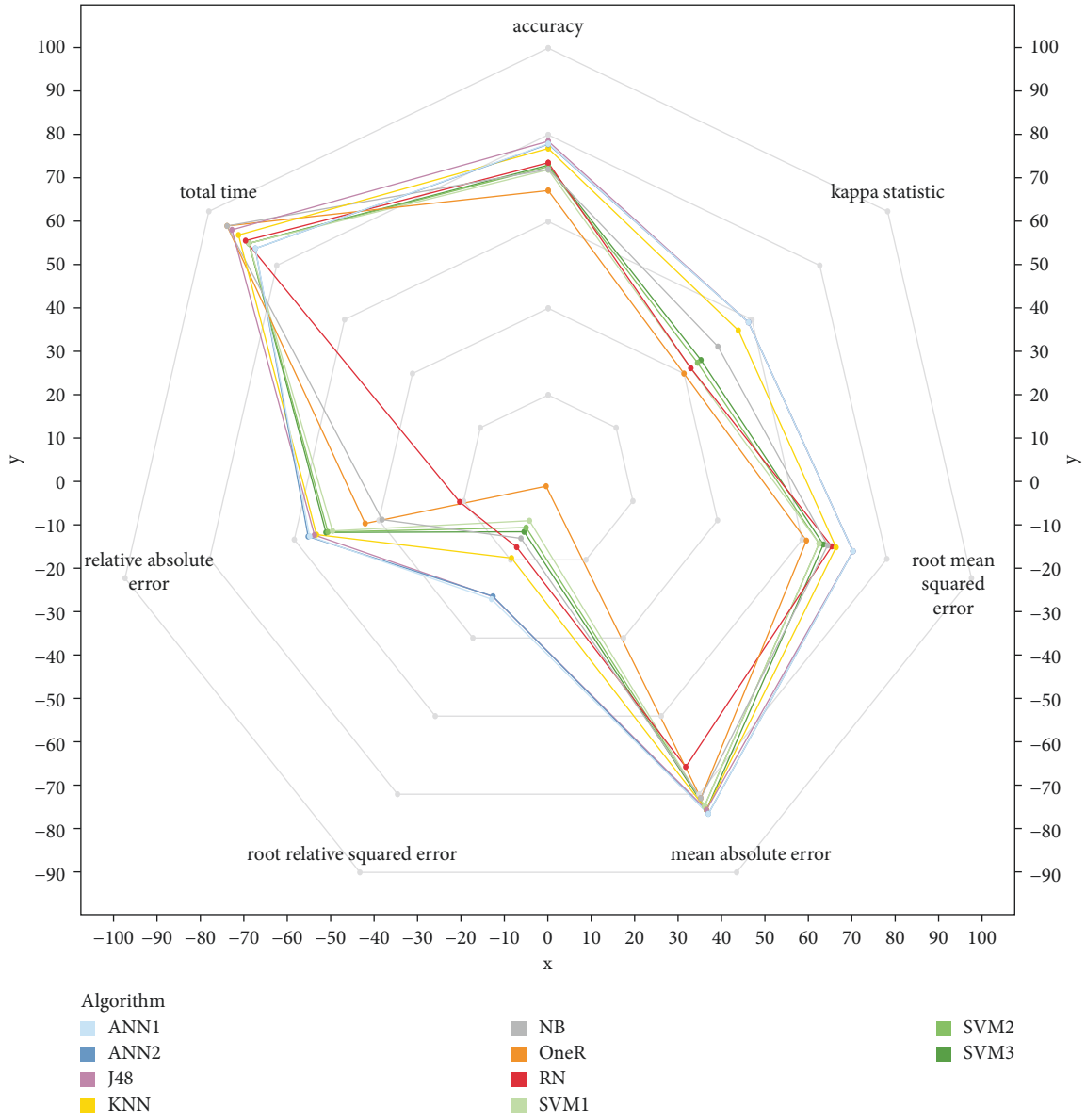


Figure 3: 6 performance metrics of 10 models on 263 datasets.

(2) As for average accuracy, the most important performance indicator, Table 5 shows that RN (73.56%) significantly outperforms OneR (67.16%) and is not statistically different from other algorithms (Figure 4(a)).

(3) For kappa statistic, RN is different from ANN1, ANN2, J48, and KNN (Figure 4(b)); however, all algorithms range from 0.41 to 0.60 (except RN); therefore, they are in the same level: moderate.

(4) About root mean squared error (RMSE) and root relative squared error (RRSE), RN outperforms OneR and lags behind ANN1, ANN2, and J48 (Figures 4(c) and 4(f)).

(5) On mean absolute error (MAE) and relative absolute error (RAE), RN is weaker than others (Figures 4(d) and 4(e)).

(6) With regard to running time, RN works faster than ANN1, ANN2, SVM1, SVM2, and SVM3, while it works slower than the remaining algorithms (Table 5).

In a word, though the proposed algorithm is simple and straightforward, it still shows the competitive performance with the above prestigious classification algorithms on core indicators.

5. Discussion

5.1. RN vs Ensemble Algorithms. As we all know, there are many powerful ensemble algorithms in the field of classification, such as random forest, random tree, and so on. However, these ensemble methods use multiple learning algorithms to achieve better predictive performance than any of the constituent learning algorithms alone. Therefore,

Table 6: The detailed experimental results on accuracy and standard deviation.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Abalone	58.87 ± 2.79	57.65 ± 2.04	60.69 ± 1.97	57.94 ± 2.50	66.63 ± 2.47	66.98 ± 2.36	66.94 ± 2.25	65.55 ± 3.38	65.62 ± 4.13	60.95 ± 1.79
Absenteeism-at-work	98.92 ± 1.53	86.08 ± 3.61	99.73 ± 0.57	83.65 ± 2.88	84.73 ± 2.99	84.73 ± 2.99	84.73 ± 2.99	97.70 ± 1.11	96.89 ± 2.21	95.27 ± 1.72
Acute-inflammation	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	79.17 ± 8.10
Acute-nephritis	100.00 ± 0.00	95.00 ± 7.03	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	91.67 ± 7.86
Adult	76.04 ± 0.06	80.41 ± 2.49	81.76 ± 3.29	78.38 ± 3.59	83.29 ± 4.11	83.41 ± 4.15	83.35 ± 4.29	80.71 ± 3.27	80.10 ± 3.07	80.34 ± 1.30
Aggregation	84.64 ± 5.38	99.87 ± 0.40	99.62 ± 0.61	99.87 ± 0.41	99.87 ± 0.41	99.75 ± 0.54	99.75 ± 0.54	99.24 ± 0.89	99.24 ± 1.07	59.64 ± 2.89
Algerianforest	89.35 ± 7.86	94.67 ± 4.73	95.50 ± 4.53	88.10 ± 6.70	58.85 ± 1.69	61.73 ± 4.28	61.73 ± 4.28	95.92 ± 3.85	96.75 ± 4.58	97.15 ± 4.75
Annealing	76.19 ± 0.13	70.43 ± 4.45	96.50 ± 2.27	89.35 ± 2.13	88.98 ± 3.15	91.48 ± 2.54	92.11 ± 2.12	91.11 ± 3.57	90.86 ± 3.39	83.71 ± 0.09
Arrhythmia	54.20 ± 1.02	60.85 ± 7.44	64.38 ± 7.08	53.55 ± 3.26	61.72 ± 2.57	65.49 ± 3.47	64.60 ± 4.20	66.38 ± 5.17	56.26 ± 13.10	59.51 ± 2.65
Au1-1000	74.10 ± 0.32	72.80 ± 1.40	77.90 ± 4.43	66.70 ± 4.72	74.10 ± 0.32	74.10 ± 0.32	74.40 ± 0.84	68.80 ± 3.79	71.00 ± 3.68	74.10 ± 0.32
Au4-2500	62.60 ± 6.11	57.30 ± 4.30	72.40 ± 5.74	63.60 ± 4.35	61.50 ± 2.68	61.50 ± 2.68	61.50 ± 2.68	64.70 ± 4.88	64.90 ± 4.89	53.60 ± 4.27
Au6-1000	23.70 ± 1.34	18.70 ± 2.95	19.40 ± 4.03	13.40 ± 3.78	24.00 ± 0.00	24.00 ± 0.00	24.00 ± 0.00	14.60 ± 3.20	13.30 ± 4.14	24.30 ± 2.21
Au6-250-drift-au6-cd1-500	22.13 ± 2.45	20.40 ± 4.75	19.33 ± 4.37	16.27 ± 4.25	22.00 ± 0.70	22.00 ± 0.70	22.00 ± 0.70	14.93 ± 2.80	15.73 ± 3.31	24.53 ± 3.45
Au6-cd1-400	28.75 ± 6.69	25.75 ± 6.35	37.00 ± 6.85	17.75 ± 7.95	27.75 ± 0.79	27.75 ± 0.79	27.75 ± 0.79	21.00 ± 6.26	20.00 ± 6.01	31.75 ± 6.57
Au7-300-drift-au7-cpd1-800	28.64 ± 4.12	34.18 ± 3.72	37.91 ± 4.35	32.00 ± 4.26	28.36 ± 1.65	27.64 ± 1.93	27.64 ± 1.93	33.45 ± 3.58	35.36 ± 4.30	24.91 ± 3.10
Au7-700	41.00 ± 6.32	42.00 ± 5.35	52.29 ± 4.00	35.57 ± 5.57	37.29 ± 2.89	36.43 ± 2.36	36.43 ± 2.36	41.29 ± 5.24	43.57 ± 3.76	44.71 ± 3.81
Au7-cpd1-500	39.40 ± 2.50	38.20 ± 5.92	53.40 ± 4.62	37.80 ± 5.69	38.20 ± 1.14	38.60 ± 1.65	38.60 ± 1.65	39.40 ± 6.40	38.00 ± 5.66	40.20 ± 4.85
Audiology-std	28.07 ± 4.62	73.04 ± 10.91	76.05 ± 7.46	74.18 ± 11.92	63.14 ± 7.94	67.81 ± 10.16	70.75 ± 8.33	84.15 ± 6.29	83.01 ± 4.43	47.39 ± 2.12
Audit-risk	90.72 ± 3.88	94.59 ± 1.98	99.87 ± 0.41	97.81 ± 1.22	98.45 ± 1.33	98.71 ± 1.35	98.84 ± 1.41	96.91 ± 2.72	96.78 ± 2.72	100.00 ± 0.00
Autism-adolescent-data	80.82 ± 7.86	98.00 ± 4.22	100.00 ± 0.00	90.45 ± 6.39	96.18 ± 4.94	95.27 ± 4.99	96.18 ± 4.94	89.45 ± 10.34	89.45 ± 10.34	100.00 ± 0.00
Autism-adult-data	83.94 ± 3.42	97.01 ± 2.64	100.00 ± 0.00	94.89 ± 3.84	99.29 ± 1.00	99.57 ± 0.96	99.43 ± 1.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Autism-child-data	91.75 ± 5.20	98.97 ± 1.67	100.00 ± 0.00	88.34 ± 3.74	99.66 ± 1.09	100.00 ± 0.00	100.00 ± 0.00	99.66 ± 1.09	99.66 ± 1.09	100.00 ± 0.00
Autos	71.67 ± 10.00	56.62 ± 12.80	82.88 ± 7.79	76.43 ± 10.04	34.64 ± 4.34	35.62 ± 5.21	35.62 ± 5.21	77.00 ± 6.32	77.02 ± 4.82	62.36 ± 9.87
Avila	97.20 ± 0.76	27.16 ± 2.24	94.01 ± 1.24	75.60 ± 2.71	67.76 ± 2.13	69.94 ± 1.87	71.38 ± 2.31	61.72 ± 1.71	61.98 ± 3.05	71.45 ± 2.52
Balance-scale	79.67 ± 2.45	90.39 ± 1.72	76.65 ± 3.77	80.97 ± 4.45	90.40 ± 1.53	92.80 ± 0.86	93.60 ± 1.32	90.70 ± 3.79	91.02 ± 3.75	56.33 ± 4.39
Balloons	65.00 ± 41.16	60.00 ± 31.62	65.00 ± 33.75	75.00 ± 35.36	70.00 ± 34.96	80.00 ± 34.96	80.00 ± 34.96	75.00 ± 26.35	75.00 ± 26.35	45.00 ± 36.89
Bank	88.48 ± 0.06	83.70 ± 1.10	88.79 ± 1.14	86.09 ± 1.35	89.54 ± 0.61	89.69 ± 0.56	89.83 ± 1.06	88.59 ± 0.89	88.39 ± 1.65	88.56 ± 0.62
Blood	76.88 ± 1.34	74.47 ± 2.97	77.81 ± 3.77	69.91 ± 4.27	78.48 ± 2.09	79.01 ± 1.54	78.48 ± 1.44	78.21 ± 2.47	78.48 ± 2.80	76.34 ± 1.66
Breast-cancer-wisc-diag	93.68 ± 3.71	92.98 ± 4.30	93.33 ± 3.94	95.96 ± 1.87	97.54 ± 2.22	97.89 ± 1.99	98.24 ± 1.17	96.13 ± 2.00	96.48 ± 2.04	89.11 ± 4.11
Breast-cancer-wisc-prog	76.29 ± 2.10	67.66 ± 11.40	74.24 ± 7.29	72.79 ± 9.05	79.32 ± 6.01	78.26 ± 8.95	77.79 ± 9.57	73.79 ± 10.91	74.26 ± 13.10	70.11 ± 7.77
Breast-cancer-wisc	96.13 ± 2.44	95.85 ± 1.72	93.85 ± 3.16	95.56 ± 2.97	97.14 ± 1.91	96.99 ± 1.96	96.71 ± 1.91	94.85 ± 3.76	95.57 ± 2.07	92.71 ± 4.54
Breast-cancer	70.64 ± 1.49	72.03 ± 5.96	71.71 ± 4.22	70.28 ± 7.73	73.83 ± 5.25	73.13 ± 5.47	72.07 ± 5.25	72.45 ± 6.54	68.90 ± 2.76	70.32 ± 5.22
Breast-tissue	68.00 ± 13.75	66.36 ± 14.92	66.45 ± 15.65	71.64 ± 13.94	58.36 ± 14.47	67.82 ± 16.52	69.82 ± 16.67	64.09 ± 8.98	63.18 ± 15.76	54.55 ± 9.71
Bupa	67.55 ± 5.03	57.17 ± 8.53	62.87 ± 9.01	61.75 ± 4.95	58.86 ± 2.27	59.15 ± 2.45	59.15 ± 2.45	66.66 ± 7.58	69.29 ± 6.28	54.24 ± 7.87
Caesarian	73.75 ± 13.76	65.00 ± 21.08	57.50 ± 17.87	61.25 ± 18.11	55.00 ± 6.45	62.50 ± 11.79	58.75 ± 16.72	56.25 ± 19.76	56.25 ± 20.62	48.75 ± 18.11
Car	70.02 ± 0.17	84.32 ± 3.29	97.57 ± 1.01	96.59 ± 1.17	97.40 ± 1.13	98.09 ± 1.16	98.32 ± 0.96	93.23 ± 1.59	93.46 ± 1.71	70.02 ± 0.17
Cardiotocography-10classes	66.65 ± 2.68	70.93 ± 2.91	83.35 ± 2.43	78.65 ± 2.97	81.23 ± 2.09	82.41 ± 2.56	83.11 ± 2.07	82.92 ± 3.08	81.42 ± 3.06	47.60 ± 2.30
Cardiotocography-3classes	77.89 ± 0.23	82.08 ± 3.32	92.90 ± 2.51	92.05 ± 1.78	91.06 ± 1.93	92.05 ± 2.09	92.19 ± 1.75	92.19 ± 1.02	92.29 ± 2.33	81.42 ± 3.18
Cervical-cancer	76.61 ± 8.44	90.36 ± 8.92	84.64 ± 13.89	88.75 ± 11.23	70.89 ± 3.16	73.75 ± 7.05	73.75 ± 7.05	92.86 ± 10.10	92.86 ± 10.10	77.86 ± 9.07
Chemicalcomposionofceramic	100.00 ± 0.00	100.00 ± 0.00	98.75 ± 3.95	100.00 ± 0.00	45.56 ± 2.34	45.56 ± 2.34	45.56 ± 2.34	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Chess-krvk	27.84 ± 2.04	30.87 ± 2.03	51.87 ± 3.06	49.95 ± 3.30	46.02 ± 2.60	48.24 ± 2.24	49.77 ± 2.93	45.42 ± 2.55	43.89 ± 2.12	25.63 ± 2.11
Chess-krvkp	76.85 ± 1.29	85.54 ± 1.84	99.31 ± 0.51	95.28 ± 1.31	98.69 ± 0.91	99.19 ± 0.63	99.19 ± 0.53	99.19 ± 0.71	99.22 ± 0.69	66.46 ± 1.44
Congressional-voting	61.15 ± 1.12	57.67 ± 7.26	59.10 ± 3.13	59.56 ± 6.13	61.61 ± 4.05	61.14 ± 5.31	60.69 ± 5.89	59.75 ± 4.88	59.55 ± 5.00	62.98 ± 2.68
Conn-bench-sonar-mines-rocks	77.02 ± 7.19	68.36 ± 9.61	71.17 ± 7.10	86.57 ± 7.01	84.17 ± 7.06	87.02 ± 7.79	88.48 ± 7.85	80.81 ± 10.93	82.21 ± 8.70	63.45 ± 10.50

Table 6: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Conn-bench-vowel-deterding	75.57 ± 2.69	65.51 ± 5.57	79.54 ± 6.12	99.25 ± 1.32	94.13 ± 1.89	97.54 ± 1.79	98.49 ± 1.49	84.65 ± 2.64	82.76 ± 3.38	35.41 ± 4.94
Connect-4	74.53 ± 0.04	68.32 ± 2.04	79.98 ± 2.16	71.37 ± 2.71	74.39 ± 0.25	76.01 ± 0.95	77.17 ± 1.74	74.21 ± 2.00	75.93 ± 1.58	74.47 ± 0.14
Connectionist	78.69 ± 8.23	67.26 ± 11.21	74.33 ± 7.93	87.05 ± 5.86	61.40 ± 8.51	69.74 ± 11.51	72.67 ± 12.19	83.52 ± 9.07	83.55 ± 8.38	62.86 ± 6.95
Contrac	48.00 ± 3.06	48.89 ± 5.47	53.36 ± 3.41	43.25 ± 3.34	54.51 ± 2.22	54.38 ± 3.54	53.77 ± 3.87	54.51 ± 3.33	55.74 ± 3.31	48.00 ± 2.03
Covid-19	75.00 ± 35.36	60.00 ± 45.95	70.00 ± 42.16	60.00 ± 45.95	60.00 ± 39.44	55.00 ± 43.78	65.00 ± 41.16	65.00 ± 41.16	65.00 ± 41.16	80.00 ± 34.96
Credit-approval	82.03 ± 4.44	76.96 ± 3.83	84.93 ± 3.82	82.17 ± 5.64	85.36 ± 5.44	85.07 ± 4.83	84.64 ± 4.59	85.22 ± 2.72	84.20 ± 4.07	85.51 ± 4.48
Crowdsource	75.77 ± 1.42	81.41 ± 2.63	88.81 ± 1.49	95.31 ± 0.96	75.34 ± 1.06	75.34 ± 1.06	75.34 ± 1.06	92.60 ± 1.88	92.46 ± 2.22	78.52 ± 2.22
Crx	83.48 ± 4.89	77.68 ± 3.15	86.09 ± 3.75	81.16 ± 4.83	55.80 ± 3.57	56.52 ± 4.27	57.83 ± 5.13	83.04 ± 4.21	83.77 ± 2.88	85.51 ± 4.48
Cryother	90.00 ± 13.30	83.33 ± 15.93	93.33 ± 7.77	90.00 ± 8.20	82.22 ± 13.04	83.33 ± 13.09	83.33 ± 13.09	87.78 ± 9.73	86.67 ± 8.76	81.11 ± 10.54
Cylinder-bands	66.22 ± 3.19	67.59 ± 5.79	72.28 ± 6.11	69.16 ± 7.22	75.78 ± 3.85	79.09 ± 4.40	81.24 ± 3.14	73.05 ± 5.28	74.03 ± 5.50	66.61 ± 4.29
Dbworld-bodies	57.38 ± 9.15	75.48 ± 15.43	80.00 ± 16.81	59.29 ± 10.50	54.52 ± 5.55	54.52 ± 5.55	63.81 ± 12.69	/	/	86.43 ± 13.28
Dbworld-bodies-stemmed	57.38 ± 9.15	76.90 ± 12.45	86.67 ± 14.29	64.05 ± 14.74	54.52 ± 5.55	59.05 ± 9.17	73.81 ± 13.23	/	/	74.29 ± 20.32
Dbworld-sub jects	54.52 ± 5.55	89.52 ± 9.93	72.62 ± 12.15	80.24 ± 13.79	54.52 ± 5.55	54.52 ± 5.55	54.52 ± 5.55	89.52 ± 13.76	87.86 ± 15.49	62.62 ± 12.35
Dbworld-sub jects-stemmed	54.52 ± 5.55	86.43 ± 13.28	75.71 ± 17.45	83.33 ± 10.47	54.52 ± 5.55	54.52 ± 5.55	54.52 ± 5.55	87.62 ± 12.28	86.19 ± 11.49	60.95 ± 10.18
Dermatology	85.53 ± 4.82	97.54 ± 2.71	96.47 ± 3.38	95.35 ± 4.26	97.55 ± 2.37	97.83 ± 2.49	97.83 ± 2.79	98.36 ± 1.41	97.82 ± 1.71	49.73 ± 2.45
Diabetes	72.52 ± 4.14	76.31 ± 5.52	73.83 ± 5.66	70.17 ± 4.69	65.11 ± 0.36	65.11 ± 0.36	65.11 ± 0.36	75.40 ± 4.66	75.39 ± 4.38	71.48 ± 5.15
Diabetic	64.38 ± 4.52	56.82 ± 2.20	64.38 ± 4.58	61.33 ± 4.75	58.21 ± 4.78	59.43 ± 6.01	59.52 ± 5.71	72.03 ± 6.10	72.46 ± 5.82	53.26 ± 5.68
Divorce	97.65 ± 3.04	97.65 ± 3.04	95.29 ± 5.41	97.65 ± 3.04	97.65 ± 3.04	97.65 ± 3.04	97.65 ± 3.04	97.65 ± 3.04	97.65 ± 3.04	95.29 ± 5.41
Dota2train	52.70 ± 0.36	54.22 ± 3.46	54.12 ± 5.39	54.43 ± 4.15	52.70 ± 0.36	53.66 ± 3.61	52.59 ± 3.29	50.44 ± 3.50	52.60 ± 2.22	55.39 ± 3.63
Dow-jones-index	65.43 ± 4.15	50.34 ± 5.13	72.50 ± 4.09	54.61 ± 6.36	52.07 ± 0.22	52.07 ± 0.22	52.07 ± 0.22	54.60 ± 5.95	55.02 ± 4.73	56.48 ± 3.08
Dry-bean-dataset	93.30 ± 1.51	90.24 ± 1.50	93.06 ± 1.13	94.62 ± 1.12	61.52 ± 2.72	61.52 ± 2.72	61.52 ± 2.72	92.53 ± 1.66	92.65 ± 1.55	68.55 ± 1.91
Early-stage-diabetes-data-upload	84.04 ± 7.08	87.12 ± 3.01	95.96 ± 2.79	98.08 ± 2.22	94.23 ± 4.15	94.81 ± 4.06	95.38 ± 3.76	96.35 ± 2.93	96.15 ± 3.27	82.31 ± 4.33
Echocardiogram	68.68 ± 7.71	79.34 ± 10.97	78.57 ± 11.42	74.01 ± 13.23	81.65 ± 8.32	80.82 ± 11.09	80.82 ± 11.09	77.14 ± 12.43	77.86 ± 13.30	85.49 ± 6.73
Ecoli	71.76 ± 4.25	86.89 ± 5.32	83.93 ± 7.54	80.37 ± 6.38	86.59 ± 4.56	86.59 ± 4.33	87.18 ± 3.51	86.00 ± 5.28	84.82 ± 3.82	66.97 ± 6.83
Eegeyesate	67.94 ± 1.22	46.77 ± 3.26	84.50 ± 1.09	83.65 ± 2.03	55.12 ± 0.03	55.16 ± 0.06	55.16 ± 0.06	54.81 ± 3.76	55.07 ± 3.68	62.60 ± 1.18
Electrical	84.91 ± 7.28	97.46 ± 4.66	100.00 ± 0.00	91.32 ± 7.32	82.67 ± 7.52	82.67 ± 6.51	83.63 ± 6.21	98.71 ± 1.67	98.40 ± 1.69	99.68 ± 1.02
Energy-y1	81.25 ± 1.04	81.77 ± 3.33	97.53 ± 1.29	75.79 ± 4.02	88.67 ± 1.39	89.84 ± 1.60	89.84 ± 2.80	88.94 ± 1.74	88.02 ± 1.82	84.50 ± 1.89
Energy-y2	89.84 ± 3.06	82.82 ± 3.31	90.23 ± 2.77	76.31 ± 3.34	89.72 ± 2.06	90.76 ± 1.55	91.28 ± 1.50	91.01 ± 2.35	91.27 ± 2.82	88.41 ± 3.09
Extentionofz-alizadehsani	71.29 ± 1.47	93.40 ± 3.49	99.68 ± 1.02	90.45 ± 5.19	71.29 ± 1.47	71.29 ± 1.47	71.29 ± 1.47	98.03 ± 2.31	98.03 ± 2.31	86.49 ± 4.71
Fertility	88.00 ± 4.22	88.00 ± 4.22	85.00 ± 7.07	83.00 ± 11.60	88.00 ± 4.22	89.00 ± 5.68	89.00 ± 7.38	90.00 ± 9.43	88.00 ± 13.17	88.00 ± 4.22
First-order	90.45 ± 0.33	31.39 ± 3.84	100.00 ± 0.00	99.41 ± 0.48	90.45 ± 0.33	95.49 ± 1.46	97.97 ± 1.09	99.87 ± 0.41	99.87 ± 0.41	90.78 ± 1.71
Flags	47.97 ± 10.07	46.34 ± 7.61	60.95 ± 8.09	45.00 ± 8.94	52.61 ± 9.41	54.24 ± 10.31	53.24 ± 13.18	46.50 ± 12.51	49.00 ± 11.93	57.26 ± 7.25
Foresttypes	93.95 ± 4.58	94.92 ± 4.16	94.97 ± 5.77	95.45 ± 3.70	30.32 ± 2.45	32.34 ± 3.64	32.34 ± 3.64	96.92 ± 3.63	96.92 ± 3.63	82.26 ± 8.50
Garments-worker-productivity	44.44 ± 2.46	26.06 ± 2.62	47.04 ± 6.19	26.15 ± 3.50	35.50 ± 2.90	36.08 ± 2.91	36.84 ± 2.47	41.19 ± 5.54	41.11 ± 4.41	32.41 ± 3.71
Gender-name-dataset	60.24 ± 1.53	58.30 ± 6.51	60.23 ± 0.15	63.60 ± 1.70	64.19 ± 1.38	64.28 ± 2.02	64.34 ± 1.81	60.32 ± 0.10	58.26 ± 6.52	60.58 ± 1.04
Gesture-a1-raw	89.18 ± 2.11	74.13 ± 2.78	91.47 ± 2.11	96.34 ± 1.12	39.95 ± 0.30	39.95 ± 0.30	39.95 ± 0.30	91.53 ± 2.23	91.36 ± 1.41	96.74 ± 1.35
Gesture-a1-va3	68.10 ± 2.47	59.27 ± 3.52	66.16 ± 4.66	74.12 ± 2.76	39.82 ± 0.22	39.82 ± 0.22	39.82 ± 0.22	70.97 ± 1.66	70.57 ± 2.31	61.56 ± 2.06
Gesture-a2-raw	88.45 ± 1.91	68.67 ± 3.16	89.25 ± 3.08	94.63 ± 3.04	39.00 ± 0.27	39.00 ± 0.27	39.00 ± 0.27	89.48 ± 1.89	89.72 ± 2.25	94.70 ± 2.62
Gesture-a2-va3	61.98 ± 2.79	43.97 ± 8.73	57.06 ± 2.60	67.70 ± 4.13	38.81 ± 0.25	38.81 ± 0.25	38.81 ± 0.25	62.30 ± 3.43	61.35 ± 4.32	53.97 ± 4.10
Gesture-a3-raw	90.68 ± 1.99	61.45 ± 3.87	92.97 ± 1.22	96.29 ± 0.99	36.10 ± 0.24	36.10 ± 0.24	36.10 ± 0.24	89.53 ± 1.45	88.77 ± 2.50	95.31 ± 1.16
Gesture-a3-va3	60.38 ± 2.32	50.22 ± 2.53	75.96 ± 3.14	89.89 ± 1.77	35.96 ± 0.23	35.96 ± 0.23	35.96 ± 0.23	65.46 ± 3.50	66.45 ± 2.70	50.93 ± 2.39
Gesture-b1-raw	91.52 ± 2.94	48.56 ± 4.91	92.73 ± 2.78	94.87 ± 2.87	38.68 ± 0.65	38.86 ± 0.79	38.86 ± 0.79	87.51 ± 4.31	89.38 ± 1.98	91.42 ± 2.24
Gesture-b1-va3	61.19 ± 5.74	34.80 ± 7.18	69.88 ± 4.24	90.56 ± 2.69	38.54 ± 0.39	38.54 ± 0.39	38.54 ± 0.39	61.84 ± 3.40	63.70 ± 4.83	42.57 ± 4.35
Gesture-b3-raw	89.40 ± 2.92	57.65 ± 4.22	91.08 ± 3.43	95.51 ± 0.88	33.08 ± 0.34	33.08 ± 0.34	33.08 ± 0.34	88.41 ± 4.10	87.64 ± 3.40	95.36 ± 2.35

Table 6: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Gesture-b3-va3	46.20 ± 3.58	43.66 ± 3.96	57.39 ± 3.97	78.03 ± 2.93	33.03 ± 0.22	33.03 ± 0.22	33.03 ± 0.22	58.10 ± 5.47	55.92 ± 5.05	40.07 ± 5.49
Gesture-c1-raw	86.41 ± 3.53	63.64 ± 2.59	89.29 ± 2.26	94.33 ± 1.68	25.74 ± 0.45	25.74 ± 0.45	25.74 ± 0.45	87.58 ± 3.36	87.13 ± 3.58	93.79 ± 2.60
Gesture-c1-va3	52.84 ± 4.08	50.50 ± 3.89	57.81 ± 3.46	70.37 ± 3.45	25.47 ± 0.36	25.47 ± 0.36	25.47 ± 0.36	64.23 ± 3.13	62.51 ± 4.15	41.46 ± 4.63
Gesture-c3-raw	88.26 ± 2.66	60.08 ± 5.35	87.77 ± 3.54	93.02 ± 2.33	27.00 ± 0.22	27.00 ± 0.22	27.00 ± 0.22	85.91 ± 3.16	83.70 ± 2.24	94.34 ± 1.34
Gesture-c3-va3	48.13 ± 3.44	45.50 ± 2.38	54.91 ± 4.00	62.39 ± 4.71	26.94 ± 0.21	26.94 ± 0.21	26.94 ± 0.21	57.41 ± 4.08	56.10 ± 5.72	42.59 ± 2.76
Glass	65.80 ± 11.01	52.27 ± 9.05	67.29 ± 8.50	70.52 ± 8.94	72.88 ± 6.56	71.90 ± 8.50	70.97 ± 8.39	67.25 ± 9.72	68.20 ± 5.31	57.08 ± 8.98
Go-track-tracks	83.60 ± 10.36	82.87 ± 4.54	87.68 ± 7.71	90.92 ± 8.57	72.02 ± 12.71	71.99 ± 11.53	71.99 ± 11.53	84.12 ± 10.34	84.74 ± 8.99	75.55 ± 13.56
Haberman-survival	73.51 ± 3.08	75.13 ± 4.07	71.86 ± 4.08	67.68 ± 6.29	72.55 ± 4.65	71.25 ± 2.91	71.25 ± 2.91	72.85 ± 6.08	73.85 ± 6.62	72.88 ± 3.00
Hayes-roth	65.88 ± 10.45	74.84 ± 13.29	80.27 ± 6.51	81.04 ± 6.53	86.32 ± 7.10	81.04 ± 6.53	80.27 ± 6.51	71.04 ± 13.20	70.27 ± 13.54	43.96 ± 5.89
Hcc-data	63.68 ± 4.52	68.97 ± 8.63	60.70 ± 7.64	63.64 ± 8.02	61.84 ± 2.26	61.84 ± 2.26	61.84 ± 2.26	64.45 ± 14.82	64.41 ± 10.82	69.15 ± 11.07
Hcvdat	87.48 ± 1.05	91.71 ± 3.64	92.04 ± 2.21	90.57 ± 1.48	86.67 ± 0.63	86.67 ± 0.63	86.67 ± 0.63	93.00 ± 2.81	92.35 ± 3.00	88.62 ± 1.84
Heart-cleveland	54.15 ± 2.28	55.84 ± 10.42	52.55 ± 8.31	54.77 ± 6.51	57.51 ± 8.53	58.83 ± 9.07	57.53 ± 9.58	54.17 ± 6.97	54.49 ± 6.43	52.54 ± 7.34
Heart-hungarian	80.33 ± 7.12	80.97 ± 8.06	78.61 ± 6.72	77.55 ± 8.10	81.72 ± 7.63	82.38 ± 6.37	83.05 ± 6.50	77.86 ± 8.83	77.83 ± 7.95	78.98 ± 9.41
Heart-switzerland	43.78 ± 8.82	41.47 ± 11.31	41.41 ± 11.01	32.69 ± 14.13	39.04 ± 6.31	39.81 ± 5.62	41.54 ± 6.48	43.97 ± 14.45	44.68 ± 11.91	32.31 ± 12.00
Heart-va	29.00 ± 9.37	33.00 ± 10.06	33.00 ± 6.32	35.00 ± 7.82	34.50 ± 9.56	32.00 ± 8.56	30.00 ± 9.43	29.50 ± 10.39	28.50 ± 10.55	27.50 ± 9.79
Heart-failure-clinical-records-dataset	69.23 ± 2.10	76.94 ± 6.31	80.61 ± 5.13	65.54 ± 6.74	67.90 ± 1.62	67.90 ± 1.62	67.90 ± 1.62	73.56 ± 4.93	77.30 ± 9.22	85.61 ± 6.32
Hepatitis	79.37 ± 2.38	83.79 ± 8.93	79.96 ± 8.81	82.50 ± 5.54	83.17 ± 6.33	84.38 ± 9.31	84.38 ± 9.31	81.96 ± 9.33	78.75 ± 11.39	74.17 ± 6.82
Hill-valley	49.82 ± 5.84	48.67 ± 2.58	50.33 ± 1.11	51.81 ± 3.48	52.16 ± 6.58	52.15 ± 7.44	53.14 ± 6.99	56.14 ± 7.60	53.14 ± 2.72	47.51 ± 4.93
Hiv1625data	93.54 ± 1.84	93.72 ± 1.70	90.95 ± 2.44	92.06 ± 1.96	92.18 ± 2.00	93.41 ± 2.12	93.72 ± 2.33	95.26 ± 2.08	93.60 ± 2.31	79.32 ± 1.71
Hiv746data	92.37 ± 3.22	90.89 ± 3.68	82.58 ± 4.76	86.74 ± 3.36	89.55 ± 3.71	90.89 ± 2.80	91.03 ± 3.27	92.50 ± 2.51	92.50 ± 3.26	80.97 ± 4.35
Horse-colic	74.00 ± 3.06	72.67 ± 5.62	87.00 ± 5.08	76.67 ± 7.54	83.67 ± 4.83	84.67 ± 5.26	85.00 ± 5.72	79.33 ± 6.05	79.67 ± 7.93	82.33 ± 5.45
Htru	97.06 ± 0.49	94.50 ± 0.50	97.84 ± 0.51	97.14 ± 0.38	91.14 ± 0.14	91.48 ± 0.19	91.38 ± 0.18	97.97 ± 0.40	97.95 ± 0.36	97.60 ± 0.40
Hypothyroid	92.29 ± 0.08	95.28 ± 0.83	99.58 ± 0.29	91.52 ± 1.55	92.60 ± 0.31	92.74 ± 0.72	92.71 ± 0.78	93.82 ± 4.15	92.92 ± 4.46	96.24 ± 0.94
Ibeacon-rssi-labeled	24.15 ± 2.44	19.86 ± 2.15	26.34 ± 3.14	37.18 ± 3.84	34.15 ± 1.70	36.55 ± 2.81	37.75 ± 3.14	27.54 ± 3.58	27.54 ± 3.99	4.51 ± 1.73
Ilpd-indian-liver	71.53 ± 1.39	55.91 ± 5.24	66.06 ± 5.25	64.83 ± 5.89	71.36 ± 0.76	70.67 ± 1.74	70.83 ± 2.51	68.95 ± 4.78	68.95 ± 6.02	66.05 ± 2.99
Image-segmentation	88.10 ± 5.14	77.62 ± 5.52	89.05 ± 7.79	87.14 ± 4.52	87.14 ± 3.21	87.14 ± 4.52	86.67 ± 4.38	90.00 ± 4.74	90.00 ± 4.74	59.05 ± 8.75
Immunotherapy	78.89 ± 3.51	75.56 ± 10.21	82.22 ± 10.73	70.00 ± 16.60	78.89 ± 3.51	78.89 ± 3.51	78.89 ± 3.51	81.11 ± 11.77	78.89 ± 8.20	84.44 ± 9.37
Impensdata	85.64 ± 1.11	90.07 ± 2.47	84.27 ± 0.30	87.54 ± 2.46	84.27 ± 0.30	84.27 ± 0.30	84.27 ± 0.30	90.28 ± 2.24	89.33 ± 2.60	84.37 ± 2.08
In-vehicle-coupon-recommendation	57.36 ± 0.20	65.59 ± 1.54	70.84 ± 1.14	63.99 ± 1.53	68.89 ± 1.32	69.42 ± 1.31	69.84 ± 1.32	70.46 ± 1.04	71.29 ± 1.09	60.53 ± 0.88
Indian	71.70 ± 0.88	55.74 ± 5.17	68.97 ± 5.22	64.12 ± 6.81	72.38 ± 1.58	71.69 ± 1.34	71.69 ± 1.34	68.26 ± 5.88	67.91 ± 7.11	65.88 ± 2.93
Ionosphere	88.31 ± 3.17	82.62 ± 5.47	91.46 ± 3.27	86.33 ± 4.59	94.31 ± 2.30	94.89 ± 3.18	95.75 ± 3.00	90.32 ± 2.39	90.90 ± 3.45	80.92 ± 7.47
Iris	81.33 ± 11.24	93.33 ± 5.44	96.00 ± 5.62	95.33 ± 5.49	96.67 ± 3.51	96.00 ± 4.66	96.67 ± 3.51	97.33 ± 3.44	97.33 ± 3.44	92.00 ± 6.13
Jain	93.53 ± 4.26	94.35 ± 4.31	99.46 ± 1.71	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	95.15 ± 3.57	95.15 ± 3.33	95.16 ± 3.32
Jsbach-chorals-harmony	88.37 ± 0.05	96.86 ± 0.55	96.52 ± 0.50	97.49 ± 0.54	96.26 ± 0.61	96.91 ± 0.45	97.19 ± 0.68	97.56 ± 0.59	97.18 ± 0.87	95.96 ± 0.62
Knowledge	61.08 ± 7.48	87.19 ± 11.04	92.35 ± 6.82	75.00 ± 8.31	83.73 ± 9.90	88.92 ± 8.89	91.83 ± 7.91	91.83 ± 5.64	92.42 ± 5.53	82.52 ± 9.66
Lasvegastripadvisorreviews	14.67 ± 3.20	16.47 ± 4.40	16.30 ± 5.05	15.68 ± 3.96	13.47 ± 3.37	11.68 ± 4.80	14.07 ± 4.82	15.69 ± 6.51	16.07 ± 5.08	14.29 ± 3.25
Leaf	58.82 ± 5.88	72.06 ± 8.57	60.88 ± 8.09	3.24 ± 3.24	17.06 ± 4.11	24.12 ± 5.15	25.59 ± 5.55	50.88 ± 7.34	48.24 ± 7.99	23.82 ± 4.89
Led-display	70.00 ± 3.62	72.00 ± 2.75	71.10 ± 3.25	70.90 ± 3.28	71.90 ± 4.09	71.70 ± 3.80	71.40 ± 3.66	71.80 ± 3.46	72.00 ± 3.89	19.60 ± 1.26
Lenses	68.33 ± 33.75	68.33 ± 33.75	78.33 ± 33.38	86.67 ± 21.94	78.33 ± 29.45	83.33 ± 22.22	83.33 ± 22.22	71.67 ± 26.12	71.67 ± 26.12	66.67 ± 30.43
Letter	71.68 ± 0.73	64.01 ± 0.74	87.91 ± 0.65	95.90 ± 0.29	94.88 ± 0.25	96.00 ± 0.30	96.51 ± 0.22	82.45 ± 1.17	81.78 ± 1.08	17.25 ± 0.37
Libras	66.94 ± 9.66	63.33 ± 9.96	69.72 ± 10.43	85.83 ± 4.98	81.11 ± 4.10	85.28 ± 4.35	86.94 ± 3.94	80.56 ± 5.24	81.11 ± 5.37	21.11 ± 6.03
Low-res-spect	66.65 ± 3.88	80.42 ± 3.96	83.24 ± 3.40	83.79 ± 4.03	89.45 ± 2.41	89.64 ± 2.72	90.58 ± 2.53	92.47 ± 3.33	91.71 ± 2.55	73.83 ± 5.30
Lung-cancer	56.67 ± 21.08	66.67 ± 20.79	40.83 ± 26.77	50.83 ± 36.53	53.33 ± 21.94	59.17 ± 21.68	51.67 ± 25.09	46.67 ± 28.11	49.17 ± 24.36	47.50 ± 17.59
Lymphography	76.24 ± 7.04	82.38 ± 8.55	74.90 ± 8.12	75.67 ± 13.05	85.86 ± 9.25	85.10 ± 9.92	83.14 ± 8.96	81.00 ± 9.18	79.62 ± 8.77	75.67 ± 8.03

Table 6: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Magic	76.45 ± 1.75	73.34 ± 2.49	82.60 ± 2.56	79.65 ± 2.34	84.75 ± 2.15	85.02 ± 1.74	85.02 ± 1.42	84.28 ± 1.79	84.02 ± 1.85	71.87 ± 4.46
Mammographic	78.87 ± 3.27	78.46 ± 3.58	82.00 ± 2.11	75.54 ± 5.00	82.73 ± 2.13	82.32 ± 2.59	82.52 ± 2.64	81.69 ± 2.69	81.17 ± 2.53	81.89 ± 1.25
Miniboone	79.08 ± 1.98	29.15 ± 1.38	85.92 ± 2.43	83.54 ± 1.59	84.54 ± 2.47	85.62 ± 2.12	86.38 ± 1.85	84.15 ± 1.71	83.85 ± 1.81	81.77 ± 1.78
Molec-biol-promoter	88.55 ± 16.48	88.45 ± 10.93	74.45 ± 10.04	72.45 ± 12.08	82.64 ± 13.45	84.64 ± 11.94	84.64 ± 11.94	78.36 ± 10.92	77.45 ± 10.10	69.73 ± 11.09
Molec-biol-splice	51.88 ± 0.17	92.35 ± 1.71	92.63 ± 1.92	63.10 ± 2.08	87.12 ± 1.97	87.27 ± 1.71	86.74 ± 1.30	84.70 ± 2.26	84.33 ± 1.77	63.32 ± 2.27
Monks-1	91.86 ± 9.63	71.54 ± 11.51	96.79 ± 6.76	70.13 ± 10.31	80.64 ± 12.11	83.08 ± 10.93	83.14 ± 9.59	88.65 ± 14.95	89.55 ± 17.20	73.27 ± 9.82
Monks-2	62.13 ± 2.93	56.18 ± 10.96	76.91 ± 10.92	61.03 ± 10.94	58.01 ± 7.39	60.92 ± 12.54	64.49 ± 12.42	73.46 ± 12.38	74.56 ± 7.84	58.60 ± 4.55
Monks-3	93.40 ± 8.60	88.53 ± 5.76	93.40 ± 8.60	74.55 ± 16.38	90.19 ± 7.56	89.29 ± 7.96	89.29 ± 7.96	84.23 ± 10.77	89.23 ± 6.89	77.88 ± 3.84
Mushroom	90.52 ± 1.06	88.84 ± 1.14	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	98.52 ± 0.48
Musk-1	84.25 ± 5.58	73.99 ± 6.97	85.11 ± 6.59	84.89 ± 4.56	90.77 ± 4.31	92.43 ± 2.86	93.69 ± 2.99	94.33 ± 2.60	93.49 ± 3.02	62.39 ± 8.95
Musk-2	87.81 ± 0.85	85.09 ± 1.67	95.42 ± 1.53	96.18 ± 0.96	96.09 ± 1.20	97.42 ± 0.95	98.06 ± 0.76	98.88 ± 0.69	98.73 ± 0.90	89.72 ± 1.26
Newdiagnosis	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	77.50 ± 6.86
Nursery	88.17 ± 1.35	89.75 ± 0.98	99.49 ± 0.27	88.98 ± 0.91	98.24 ± 0.22	98.67 ± 0.21	98.94 ± 0.20	95.73 ± 0.88	94.77 ± 1.30	70.97 ± 1.03
Obesitydataset-raw-and-data-sinthetic	83.04 ± 2.29	67.41 ± 2.79	93.75 ± 1.94	82.05 ± 2.48	89.72 ± 2.33	91.71 ± 2.24	92.47 ± 2.22	94.37 ± 2.75	94.22 ± 1.46	67.08 ± 3.58
Obs-network-dataset-2-aug27	96.00 ± 1.70	71.82 ± 5.31	99.63 ± 1.17	99.07 ± 0.62	97.68 ± 0.65	97.68 ± 0.65	97.68 ± 0.65	96.56 ± 1.53	98.05 ± 1.19	90.69 ± 2.07
Occupancy-data	96.13 ± 0.92	95.53 ± 1.02	98.61 ± 0.81	99.14 ± 0.50	80.94 ± 2.01	82.10 ± 2.31	82.10 ± 2.31	97.94 ± 0.87	98.01 ± 0.90	98.69 ± 0.77
Occupancy-data2	98.41 ± 0.43	96.03 ± 0.55	99.44 ± 0.16	99.47 ± 0.23	83.55 ± 0.50	85.65 ± 0.59	85.65 ± 0.59	99.43 ± 0.21	99.42 ± 0.15	99.37 ± 0.16
Occupancy-data3	98.40 ± 0.63	97.70 ± 0.55	99.51 ± 0.10	99.46 ± 0.23	82.91 ± 0.53	84.51 ± 0.61	84.53 ± 0.61	98.76 ± 0.37	98.77 ± 0.30	99.37 ± 0.18
Old	88.60 ± 1.80	94.05 ± 0.97	94.99 ± 1.72	97.68 ± 0.82	95.07 ± 0.93	95.19 ± 1.27	95.48 ± 1.19	96.95 ± 0.97	96.86 ± 1.12	88.68 ± 1.62
Online-shoppers-intention	84.53 ± 0.03	81.60 ± 1.30	89.53 ± 0.53	81.53 ± 0.57	84.50 ± 0.06	84.45 ± 0.13	84.40 ± 0.15	89.38 ± 0.85	88.86 ± 1.21	88.23 ± 0.98
Oocytes-merluccius-nucleus-4d	72.02 ± 2.48	59.98 ± 4.09	75.64 ± 2.43	72.22 ± 3.90	77.30 ± 2.90	80.13 ± 2.51	81.01 ± 3.02	81.41 ± 2.04	80.62 ± 4.38	66.83 ± 5.20
Oocytes-merluccius-states-2f	88.55 ± 3.21	84.64 ± 4.04	90.51 ± 3.40	91.09 ± 2.84	91.68 ± 2.59	92.07 ± 2.10	92.76 ± 1.97	91.78 ± 1.69	91.29 ± 2.76	81.89 ± 4.69
Oocytes-trisopterus-nucleus-2f	g64.46 ± 5.21	53.07 ± 5.55	72.16 ± 6.85	74.34 ± 5.80	81.15 ± 3.67	83.78 ± 3.41	83.34 ± 2.76	83.11 ± 3.97	82.12 ± 4.25	59.43 ± 5.98
Oocytes-trisopterus-states-5b	g83.13 ± 4.78	75.78 ± 4.23	88.60 ± 2.55	90.90 ± 3.69	91.78 ± 3.40	92.33 ± 2.74	92.77 ± 2.32	93.32 ± 2.68	93.32 ± 2.97	80.04 ± 2.74
Optdigits	88.83 ± 1.19	91.33 ± 1.01	90.69 ± 1.44	98.61 ± 0.64	73.13 ± 1.91	74.88 ± 2.08	74.88 ± 2.08	98.26 ± 0.71	98.43 ± 0.54	27.03 ± 1.23
Optical	88.47 ± 1.57	91.68 ± 1.31	89.72 ± 1.39	98.46 ± 0.57	98.59 ± 0.55	98.74 ± 0.60	98.72 ± 0.58	98.27 ± 0.54	98.20 ± 0.75	27.70 ± 1.30
Ozone	97.12 ± 0.19	71.10 ± 3.68	95.78 ± 0.53	95.11 ± 0.67	97.12 ± 0.19	97.12 ± 0.19	97.12 ± 0.19	96.22 ± 0.97	96.65 ± 0.89	96.85 ± 0.45
Page-blocks	92.84 ± 0.89	89.99 ± 1.43	96.88 ± 0.41	96.02 ± 0.60	96.18 ± 0.75	96.44 ± 0.75	96.47 ± 0.76	96.22 ± 0.70	96.20 ± 0.55	93.75 ± 0.60
Parkingbirmingham	81.18 ± 1.94	76.17 ± 2.07	100.00 ± 0.00	97.45 ± 0.67	58.69 ± 3.78	61.30 ± 3.91	61.30 ± 3.91	86.42 ± 1.54	87.88 ± 1.56	100.00 ± 0.00
Parkinsons	84.63 ± 5.40	69.34 ± 11.51	80.50 ± 3.33	96.39 ± 2.49	87.71 ± 6.39	89.79 ± 6.29	91.34 ± 5.80	90.63 ± 7.79	89.66 ± 8.93	86.18 ± 7.59
Pasture	71.67 ± 24.91	74.17 ± 23.72	78.33 ± 20.49	71.67 ± 27.55	28.33 ± 4.30	28.33 ± 4.30	28.33 ± 4.30	76.67 ± 25.09	74.17 ± 26.48	63.33 ± 26.41
Pbc	75.35 ± 4.83	77.72 ± 7.42	77.24 ± 8.31	59.58 ± 8.77	61.48 ± 0.79	61.48 ± 0.79	61.48 ± 0.79	71.06 ± 6.04	73.67 ± 4.02	72.25 ± 9.13
Pen	87.26 ± 0.61	85.75 ± 1.18	96.56 ± 0.53	99.36 ± 0.17	13.49 ± 0.86	14.47 ± 0.95	14.47 ± 0.95	94.62 ± 0.63	94.40 ± 0.43	38.95 ± 1.54
Pendigits	86.55 ± 1.02	87.99 ± 1.46	96.14 ± 0.75	99.43 ± 0.24	99.61 ± 0.19	99.67 ± 0.16	99.65 ± 0.17	95.17 ± 0.67	95.05 ± 0.54	39.74 ± 1.98
Pharynx	76.45 ± 2.19	68.24 ± 10.48	76.45 ± 2.19	64.08 ± 10.16	75.42 ± 8.61	73.29 ± 10.60	70.26 ± 10.68	71.79 ± 3.64	70.95 ± 16.41	27.13 ± 5.60
Phishingwebsites	85.07 ± 0.99	92.98 ± 0.64	95.88 ± 0.44	97.18 ± 0.45	94.50 ± 0.52	95.00 ± 0.50	95.26 ± 0.46	96.78 ± 0.38	96.83 ± 0.48	88.89 ± 0.56
Pima	72.00 ± 3.77	76.31 ± 5.55	73.83 ± 5.66	70.17 ± 4.69	75.66 ± 4.08	75.13 ± 4.01	75.00 ± 3.43	75.40 ± 4.66	75.39 ± 4.38	71.48 ± 5.15
Pittsburg-bridges-rel-l	69.09 ± 8.02	66.91 ± 11.95	63.00 ± 14.10	71.09 ± 16.91	67.18 ± 12.34	68.09 ± 12.39	72.82 ± 11.10	61.36 ± 14.27	66.18 ± 10.75	71.00 ± 10.59
Pittsburg-bridges-span	59.78 ± 8.98	67.33 ± 13.93	60.78 ± 13.17	55.00 ± 18.05	65.11 ± 14.63	64.00 ± 16.18	64.00 ± 16.18	66.22 ± 9.91	68.44 ± 13.31	54.67 ± 14.02
Pittsburg-bridges-t-or-d	86.36 ± 4.73	86.45 ± 9.83	83.55 ± 10.78	83.55 ± 11.76	86.45 ± 7.74	88.45 ± 8.44	89.45 ± 7.90	85.64 ± 12.49	83.64 ± 14.10	85.27 ± 9.69
Pittsburg-bridges-type	55.91 ± 11.28	55.91 ± 11.62	61.73 ± 9.31	57.82 ± 11.76	53.91 ± 12.05	61.27 ± 17.65	62.45 ± 13.32	59.64 ± 17.82	59.64 ± 17.30	56.09 ± 5.40
Pittsburg-bridgesmaterial	84.00 ± 7.75	84.00 ± 7.50	87.73 ± 8.07	83.73 ± 11.42	85.91 ± 6.39	85.00 ± 6.24	84.00 ± 9.64	83.18 ± 11.03	79.36 ± 10.14	86.82 ± 4.74
Planning	72.02 ± 2.57	68.13 ± 6.28	71.46 ± 1.60	65.94 ± 9.32	71.46 ± 1.60	70.32 ± 5.96	65.91 ± 6.93	61.49 ± 7.03	58.71 ± 9.93	62.05 ± 8.60
Plant-margin	63.13 ± 4.31	84.69 ± 2.59	48.06 ± 4.50	74.00 ± 2.78	82.81 ± 2.27	84.75 ± 2.80	85.13 ± 2.82	83.06 ± 2.23	83.63 ± 3.24	7.69 ± 2.17

Table 6: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Plant-shape	49.81 ± 4.42	53.19 ± 2.88	46.13 ± 3.40	64.69 ± 3.35	48.94 ± 1.62	56.75 ± 3.13	60.06 ± 3.35	64.75 ± 3.79	65.00 ± 3.35	8.56 ± 1.62
Plant-texture	62.85 ± 2.93	74.42 ± 3.13	51.97 ± 3.48	80.11 ± 2.84	83.30 ± 2.50	85.55 ± 2.29	86.06 ± 2.35	83.80 ± 1.74	83.11 ± 1.91	3.44 ± 0.90
Poker-hand-training-true	49.95 ± 0.02	49.95 ± 0.02	55.91 ± 2.04	47.03 ± 0.86	58.71 ± 1.00	57.57 ± 0.93	56.45 ± 1.11	53.97 ± 0.61	51.90 ± 2.03	49.95 ± 0.02
Post-operative	71.11 ± 5.74	65.56 ± 6.31	71.11 ± 5.74	61.11 ± 10.80	71.11 ± 5.74	68.89 ± 7.03	66.67 ± 9.07	56.67 ± 13.30	56.67 ± 16.10	68.89 ± 7.03
Primary-tumor	26.06 ± 2.12	43.03 ± 10.18	43.33 ± 9.04	35.15 ± 8.35	46.67 ± 10.90	44.85 ± 10.57	43.94 ± 11.97	40.91 ± 7.99	41.21 ± 8.48	27.88 ± 2.78
Qsarbioconcentration	74.84 ± 0.63	73.69 ± 1.80	73.29 ± 3.81	61.62 ± 5.75	74.84 ± 0.70	73.30 ± 1.50	70.60 ± 2.89	70.99 ± 4.07	71.63 ± 3.05	72.79 ± 3.27
Qsarbiodegradation	69.95 ± 2.16	75.83 ± 3.23	83.51 ± 3.46	83.60 ± 2.24	85.88 ± 2.62	86.25 ± 2.17	85.59 ± 2.49	86.92 ± 2.30	87.21 ± 2.46	76.69 ± 3.95
Qualitative-bankruptcy	99.20 ± 2.53	99.20 ± 2.53	98.00 ± 2.83	100.00 ± 0.00	98.80 ± 2.70	98.80 ± 2.70	99.60 ± 1.26	99.20 ± 2.53	99.20 ± 2.53	98.40 ± 2.80
Ringnorm	79.14 ± 1.43	98.64 ± 0.47	91.61 ± 0.27	75.16 ± 1.58	98.59 ± 0.41	98.47 ± 0.41	98.35 ± 0.48	91.95 ± 1.47	91.70 ± 0.86	64.43 ± 0.97
Risk-factors-cervical-cancer	93.59 ± 0.60	88.69 ± 1.91	95.10 ± 1.97	94.40 ± 2.06	93.59 ± 0.60	93.47 ± 0.80	93.94 ± 1.62	94.76 ± 2.00	94.52 ± 2.27	96.15 ± 1.91
Robotnavigation	91.81 ± 1.00	52.51 ± 2.01	99.60 ± 0.23	88.25 ± 1.22	90.03 ± 1.14	91.18 ± 0.98	91.99 ± 0.77	88.21 ± 1.86	88.20 ± 2.44	75.37 ± 1.90
Sapfile	48.85 ± 11.47	54.18 ± 10.31	45.71 ± 14.95	41.98 ± 8.97	48.85 ± 8.89	49.67 ± 6.83	49.67 ± 8.54	48.08 ± 9.46	47.31 ± 12.30	41.92 ± 12.36
Sat	85.10 ± 0.87	79.60 ± 1.23	86.03 ± 0.98	90.55 ± 0.79	26.14 ± 0.53	27.32 ± 0.47	27.32 ± 0.47	89.71 ± 1.60	88.98 ± 1.11	59.75 ± 1.30
Satelite	85.07 ± 0.82	79.56 ± 1.10	86.34 ± 0.65	90.58 ± 1.11	26.12 ± 0.57	27.40 ± 0.71	27.40 ± 0.71	89.82 ± 0.66	89.26 ± 0.87	59.97 ± 1.65
Scadi	58.57 ± 8.11	82.86 ± 9.04	80.00 ± 13.80	80.00 ± 13.80	75.71 ± 9.64	78.57 ± 13.88	81.43 ± 15.13	80.00 ± 9.99	80.00 ± 9.99	58.57 ± 10.54
Schillingdata	86.86 ± 0.24	93.86 ± 0.74	86.74 ± 0.15	88.45 ± 1.00	86.74 ± 0.15	89.00 ± 1.03	91.63 ± 0.66	92.67 ± 2.08	91.01 ± 3.38	86.58 ± 0.44
Seeds	86.67 ± 6.66	91.43 ± 4.38	91.90 ± 5.96	94.29 ± 2.01	93.81 ± 3.92	93.81 ± 3.92	93.81 ± 3.92	95.24 ± 3.17	96.19 ± 3.01	82.38 ± 10.05
Segment	90.30 ± 2.26	80.22 ± 1.89	96.93 ± 1.09	97.14 ± 0.62	65.37 ± 2.88	67.01 ± 2.40	66.97 ± 2.47	96.06 ± 1.20	96.28 ± 0.89	63.85 ± 2.78
Seismic-bumps	93.42 ± 0.01	86.73 ± 1.89	93.34 ± 0.24	89.39 ± 1.46	93.42 ± 0.01	93.42 ± 0.01	93.42 ± 0.01	92.45 ± 0.93	92.30 ± 0.92	92.96 ± 0.55
Semeion	78.02 ± 3.36	85.94 ± 2.72	75.83 ± 2.39	91.46 ± 1.92	95.60 ± 1.07	96.17 ± 0.69	96.11 ± 0.72	92.65 ± 1.57	93.09 ± 1.66	19.59 ± 0.47
Setapprocesst1	68.33 ± 8.70	58.57 ± 22.11	62.86 ± 20.40	65.24 ± 10.72	63.81 ± 8.38	68.57 ± 10.64	66.90 ± 9.29	64.76 ± 20.48	65.00 ± 19.92	72.14 ± 23.19
Setapprocesst10	66.43 ± 5.51	44.64 ± 15.99	58.21 ± 14.12	60.54 ± 16.51	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	51.25 ± 27.07	51.25 ± 21.95	62.14 ± 8.07
Setapprocesst11	66.43 ± 5.51	52.50 ± 24.67	65.71 ± 18.55	58.04 ± 16.28	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	61.43 ± 15.71	64.11 ± 20.11	56.96 ± 10.60
Setapprocesst2	67.86 ± 8.16	66.79 ± 15.27	72.14 ± 19.71	62.68 ± 21.57	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	62.68 ± 20.16	65.18 ± 20.81	81.25 ± 16.71
Setapprocesst3	66.43 ± 5.51	65.18 ± 12.69	60.54 ± 15.76	66.25 ± 13.40	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	66.25 ± 13.40	66.25 ± 9.43	53.04 ± 16.45
Setapprocesst4	71.43 ± 6.64	47.62 ± 26.15	55.24 ± 15.22	60.24 ± 18.58	71.43 ± 6.64	71.43 ± 6.64	71.43 ± 6.64	55.00 ± 21.89	56.90 ± 17.37	72.62 ± 11.68
Setapprocesst5	69.29 ± 7.28	45.89 ± 16.52	53.39 ± 18.00	59.64 ± 17.60	65.18 ± 7.54	63.93 ± 8.94	63.93 ± 8.94	59.82 ± 13.58	62.50 ± 14.68	65.71 ± 24.88
Setapprocesst6	66.43 ± 5.51	67.50 ± 16.68	64.82 ± 13.15	64.29 ± 14.51	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	62.68 ± 17.46	62.68 ± 14.64	60.18 ± 20.15
Setapprocesst7	67.86 ± 8.16	74.82 ± 14.86	60.71 ± 19.94	64.11 ± 15.09	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	70.54 ± 21.15	71.96 ± 18.29	52.86 ± 15.18
Setapprocesst8	66.43 ± 5.51	68.93 ± 14.02	70.18 ± 15.13	63.21 ± 15.59	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	66.61 ± 16.37	65.36 ± 18.16	49.46 ± 11.45
Setapprocesst9	66.43 ± 5.51	64.64 ± 16.62	69.29 ± 16.49	58.39 ± 9.93	66.43 ± 5.51	66.43 ± 5.51	66.43 ± 5.51	70.18 ± 20.26	67.68 ± 17.22	50.71 ± 16.54
Shillbiddingdataset	90.95 ± 0.43	96.71 ± 0.43	99.62 ± 0.15	99.22 ± 0.28	99.54 ± 0.17	99.56 ± 0.21	99.65 ± 0.19	99.68 ± 0.15	99.72 ± 0.19	97.28 ± 0.54
Shuttle-landing-control	95.00 ± 15.81	80.00 ± 34.96	95.00 ± 15.81	95.00 ± 15.81	95.00 ± 15.81	95.00 ± 15.81	95.00 ± 15.81	85.00 ± 33.75	80.00 ± 34.96	85.00 ± 33.75
Somervillehappinesssurvey2015	57.33 ± 12.50	59.38 ± 9.99	64.29 ± 17.09	58.95 ± 12.80	60.05 ± 10.86	58.71 ± 10.31	58.76 ± 11.69	58.86 ± 9.38	59.57 ± 11.65	65.67 ± 15.87
Sonar	76.55 ± 7.06	67.88 ± 9.29	71.17 ± 7.10	86.57 ± 7.01	65.90 ± 5.43	70.26 ± 9.01	73.60 ± 9.54	82.29 ± 10.70	81.76 ± 9.68	62.50 ± 11.07
Soybean	69.10 ± 6.56	91.86 ± 5.60	87.26 ± 5.16	87.62 ± 4.79	90.22 ± 6.00	92.17 ± 4.45	91.53 ± 4.69	89.88 ± 6.85	90.84 ± 6.47	30.63 ± 2.45
Spambase	77.42 ± 1.19	79.11 ± 1.78	92.96 ± 1.19	90.78 ± 1.10	93.39 ± 0.93	93.50 ± 1.10	93.41 ± 0.92	91.20 ± 1.27	89.11 ± 6.86	78.35 ± 1.99
Speaker-accent	51.36 ± 2.52	58.07 ± 10.22	69.92 ± 9.58	80.56 ± 8.92	58.36 ± 4.00	62.31 ± 4.31	62.31 ± 4.31	79.64 ± 8.20	75.40 ± 7.29	50.45 ± 4.64
Spect	70.89 ± 8.34	72.14 ± 17.46	70.89 ± 13.18	55.71 ± 15.72	64.64 ± 11.19	64.46 ± 11.74	63.21 ± 9.46	55.71 ± 17.79	56.96 ± 14.57	72.14 ± 9.82
Spectf	68.75 ± 12.15	75.00 ± 11.79	73.75 ± 12.43	67.50 ± 18.82	67.50 ± 6.45	67.50 ± 6.45	76.25 ± 13.76	71.25 ± 14.49	72.50 ± 15.37	71.25 ± 8.44
Statlog-australian-credit	67.68 ± 0.70	59.28 ± 5.53	66.96 ± 7.13	55.80 ± 3.29	67.54 ± 1.83	64.35 ± 6.38	63.91 ± 5.61	62.75 ± 5.59	61.01 ± 4.50	65.22 ± 6.34
Statlog-german-credit	70.00 ± 0.00	75.60 ± 5.21	73.90 ± 4.68	67.80 ± 3.94	76.10 ± 4.12	77.30 ± 3.74	77.30 ± 4.42	70.90 ± 3.93	72.20 ± 3.36	70.90 ± 2.28
Statlog-heart	83.70 ± 7.45	83.33 ± 7.25	76.30 ± 9.75	75.19 ± 8.56	82.96 ± 5.30	80.00 ± 8.41	79.63 ± 9.60	78.15 ± 7.08	80.00 ± 8.76	71.11 ± 6.25
Statlog-image	90.22 ± 2.15	80.09 ± 2.02	96.93 ± 1.09	97.14 ± 0.62	94.29 ± 0.88	95.15 ± 0.95	95.45 ± 0.90	96.10 ± 0.91	96.15 ± 1.05	63.68 ± 2.46

Table 6: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Statlog-landsat	85.95 ± 1.49	79.55 ± 2.20	86.13 ± 1.81	90.15 ± 1.70	89.47 ± 1.59	90.19 ± 1.38	90.26 ± 1.61	89.00 ± 1.26	88.88 ± 1.74	60.45 ± 1.68
Statlog-shuttle	84.19 ± 0.43	91.64 ± 0.89	99.96 ± 0.03	99.94 ± 0.04	99.75 ± 0.05	99.80 ± 0.05	99.80 ± 0.04	99.69 ± 0.04	99.71 ± 0.07	94.69 ± 0.30
Statlog-vehicle	66.55 ± 2.71	44.80 ± 3.40	72.23 ± 5.93	69.86 ± 4.47	76.72 ± 3.88	80.26 ± 2.60	80.50 ± 2.36	82.28 ± 3.15	80.75 ± 4.62	51.19 ± 5.09
Steel-plates	61.62 ± 2.95	60.28 ± 3.84	76.04 ± 3.30	71.97 ± 2.73	75.53 ± 2.77	76.97 ± 2.27	76.87 ± 2.92	72.70 ± 2.86	72.65 ± 3.17	48.17 ± 3.88
Synthetic-control	89.00 ± 2.74	94.67 ± 3.50	91.67 ± 3.51	96.50 ± 2.14	99.33 ± 0.86	99.50 ± 0.81	99.50 ± 0.81	99.33 ± 0.86	99.33 ± 0.86	57.00 ± 3.83
Teaching	50.25 ± 11.28	54.21 ± 8.37	59.58 ± 11.57	62.29 ± 13.54	56.17 ± 14.70	61.50 ± 12.68	62.17 ± 12.37	54.29 ± 11.11	52.29 ± 9.41	46.38 ± 10.93
Thoraricsurgery	85.11 ± 0.00	78.51 ± 9.27	84.47 ± 1.03	77.23 ± 5.94	85.11 ± 0.00	84.47 ± 1.03	84.47 ± 1.03	79.15 ± 4.23	79.79 ± 5.42	83.40 ± 1.68
Thyroid	85.19 ± 7.42	96.73 ± 3.84	92.06 ± 6.32	97.21 ± 2.40	75.87 ± 5.01	77.23 ± 4.51	77.23 ± 4.51	96.75 ± 3.10	96.75 ± 3.10	90.69 ± 8.54
Thyroid-train	92.47 ± 0.13	95.63 ± 1.10	99.71 ± 0.32	92.13 ± 1.32	95.10 ± 0.68	95.68 ± 0.60	96.26 ± 0.63	96.58 ± 1.05	96.00 ± 1.30	96.47 ± 0.78
Tic-tac-toe	65.34 ± 0.43	69.62 ± 3.56	93.84 ± 2.44	100.00 ± 0.00	98.64 ± 1.40	99.27 ± 1.00	99.69 ± 0.71	97.81 ± 1.04	97.91 ± 1.21	69.93 ± 2.39
Titanic	78.05 ± 2.46	75.37 ± 2.86	79.05 ± 1.78	79.05 ± 1.78	78.46 ± 2.19	78.92 ± 2.01	78.92 ± 2.01	78.15 ± 2.26	77.42 ± 2.19	77.60 ± 2.02
Trains	70.00 ± 48.30	60.00 ± 51.64	70.00 ± 48.30	60.00 ± 51.64	60.00 ± 51.64	70.00 ± 48.30	70.00 ± 48.30	80.00 ± 42.16	80.00 ± 42.16	0.00 ± 0.00
Transfusion	76.88 ± 1.34	75.40 ± 3.44	77.81 ± 3.77	70.45 ± 4.45	75.27 ± 2.89	72.99 ± 3.06	71.92 ± 3.72	78.21 ± 2.47	78.48 ± 2.80	76.34 ± 1.66
Trial	94.08 ± 2.90	90.34 ± 3.48	100.00 ± 0.00	99.74 ± 0.82	99.10 ± 0.87	99.36 ± 0.68	99.36 ± 0.68	99.23 ± 0.90	99.36 ± 0.91	100.00 ± 0.00
Turkiye-student-evaluation	61.87 ± 0.05	86.99 ± 1.52	99.97 ± 0.07	90.41 ± 1.58	95.50 ± 0.52	96.27 ± 0.59	96.27 ± 0.59	100.00 ± 0.00	100.00 ± 0.00	99.97 ± 0.07
Unbalanced	98.60 ± 0.49	90.77 ± 4.44	98.60 ± 0.49	97.66 ± 1.57	98.60 ± 0.49	98.37 ± 0.60	98.13 ± 0.81	98.13 ± 1.12	97.90 ± 1.08	98.60 ± 0.49
Urbanlandcover	80.18 ± 10.26	78.49 ± 9.22	79.08 ± 7.80	76.65 ± 11.07	17.28 ± 1.99	17.28 ± 1.99	17.28 ± 1.99	77.21 ± 12.43	77.21 ± 12.43	49.23 ± 14.54
Userknowledgemodeling	63.91 ± 7.90	89.34 ± 5.79	94.02 ± 3.41	80.58 ± 10.14	77.74 ± 6.42	85.29 ± 4.31	85.69 ± 3.93	91.29 ± 3.59	93.25 ± 4.58	85.31 ± 7.80
Vehicle	69.27 ± 4.29	44.80 ± 4.72	72.47 ± 5.87	69.86 ± 4.47	30.50 ± 2.54	31.32 ± 3.07	31.32 ± 3.07	*81.70 ± 3.82*	81.21 ± 4.43	51.90 ± 5.04
Vertebral-column-2classes	74.52 ± 8.93	77.74 ± 7.04	81.61 ± 7.91	81.94 ± 5.09	*85.48 ± 5.32*	84.52 ± 6.23	84.84 ± 5.70	84.52 ± 6.77	84.19 ± 6.17	73.87 ± 7.67
Vertebral-column-3classes	70.00 ± 10.32	83.23 ± 5.85	82.26 ± 6.13	78.71 ± 4.86	83.87 ± 5.89	84.84 ± 6.98	84.19 ± 6.71	*85.16 ± 4.86*	84.52 ± 6.04	75.16 ± 6.98
Veteran	71.48 ± 2.70	72.09 ± 8.96	72.20 ± 6.80	60.66 ± 8.86	70.77 ± 1.06	70.77 ± 1.06	70.77 ± 1.06	60.60 ± 14.50	67.14 ± 15.64	*73.57 ± 7.61*
Vowel	62.42 ± 3.43	63.74 ± 4.43	81.52 ± 4.72	*99.29 ± 0.83*	85.05 ± 2.60	91.11 ± 2.01	92.93 ± 2.08	92.83 ± 3.45	93.54 ± 2.39	32.02 ± 4.21
Wall-following	91.86 ± 1.22	52.47 ± 1.38	*99.65 ± 0.25*	88.18 ± 1.21	89.00 ± 1.41	90.60 ± 1.58	91.35 ± 1.30	87.48 ± 1.40	87.99 ± 1.95	75.64 ± 1.43
Waveform-noise	67.18 ± 2.20	80.00 ± 1.93	75.16 ± 1.37	73.62 ± 1.27	*86.14 ± 1.51*	85.38 ± 1.28	84.90 ± 1.41	83.54 ± 1.64	83.08 ± 1.43	53.72 ± 2.75
Waveform	70.94 ± 1.45	81.02 ± 1.38	75.88 ± 1.31	76.90 ± 2.01	*86.42 ± 1.36*	85.66 ± 1.38	85.46 ± 1.43	83.84 ± 1.15	83.16 ± 1.28	52.54 ± 1.64
Wbc	96.92 ± 2.02	97.51 ± 1.39	92.98 ± 2.02	95.76 ± 2.52	97.66 ± 1.58	97.66 ± 1.58	*97.80 ± 1.73*	95.76 ± 1.59	96.19 ± 1.85	91.06 ± 2.91
Wdbc	93.68 ± 3.71	92.98 ± 4.30	93.33 ± 3.94	95.96 ± 1.87	62.74 ± 0.73	62.74 ± 0.73	62.74 ± 0.73	*96.66 ± 1.76*	95.96 ± 2.35	89.81 ± 3.17
Weathernominal	65.00 ± 41.16	60.00 ± 39.44	55.00 ± 43.78	65.00 ± 41.16	70.00 ± 34.96	60.00 ± 39.44	*75.00 ± 35.36*	*75.00 ± 35.36*	*75.00 ± 35.36*	45.00 ± 43.78
Weathernumeric	55.00 ± 43.78	70.00 ± 34.96	70.00 ± 42.16	*80.00 ± 34.96*	70.00 ± 34.96	50.00 ± 47.14	50.00 ± 47.14	75.00 ± 42.49	75.00 ± 42.49	45.00 ± 43.78
Website-phishingdata	82.41 ± 2.24	84.11 ± 1.90	*90.76 ± 2.08*	88.32 ± 2.77	85.66 ± 1.27	86.18 ± 1.20	87.65 ± 2.05	88.77 ± 1.94	89.80 ± 2.46	81.74 ± 1.63
Wholesalecustomersdata	89.55 ± 2.20	89.09 ± 3.83	90.23 ± 3.04	87.95 ± 4.80	67.73 ± 0.96	67.73 ± 0.96	67.73 ± 0.96	*91.59 ± 3.40*	90.91 ± 4.01	90.45 ± 3.68
Wifi-localization	92.60 ± 1.68	98.20 ± 0.71	97.15 ± 0.63	*98.30 ± 0.89*	79.05 ± 2.49	80.45 ± 2.58	80.45 ± 2.58	97.90 ± 0.39	97.95 ± 0.69	79.00 ± 4.45
Wilt	94.61 ± 0.07	89.17 ± 1.82	*98.20 ± 0.46*	94.88 ± 0.58	94.77 ± 0.14	94.77 ± 0.14	94.77 ± 0.14	97.85 ± 0.87	98.12 ± 0.58	94.46 ± 0.21
Wine-quality-red	61.98 ± 3.57	55.10 ± 4.59	61.16 ± 2.98	*64.79 ± 2.73*	62.79 ± 2.48	63.17 ± 2.68	63.29 ± 2.70	60.48 ± 3.81	60.60 ± 3.07	54.66 ± 3.11
Wine-quality-white	57.17 ± 1.97	44.10 ± 1.20	58.35 ± 2.80	*65.40 ± 2.50*	57.00 ± 2.28	57.86 ± 2.47	58.11 ± 2.64	55.25 ± 2.32	54.61 ± 1.46	45.75 ± 2.15
Wine	96.11 ± 5.89	96.63 ± 5.38	93.86 ± 5.52	94.97 ± 4.11	*98.89 ± 2.34*	98.30 ± 2.74	98.30 ± 2.74	97.19 ± 3.96	97.19 ± 3.96	76.96 ± 9.24
Yamilnaduelectricty	19.88 ± 1.29	6.65 ± 0.37	14.06 ± 1.09	*21.21 ± 1.31*	6.65 ± 0.55	6.82 ± 0.56	6.70 ± 0.61	6.40 ± 0.72	6.40 ± 0.69	7.45 ± 1.12
Yeast	46.83 ± 2.58	57.81 ± 2.65	55.99 ± 4.85	52.29 ± 2.39	60.24 ± 4.08	60.31 ± 4.24	*60.44 ± 4.50*	59.43 ± 3.49	58.82 ± 3.39	40.02 ± 3.39
Youtobe-kabita-preprocessing	33.18 ± 2.32	26.61 ± 1.02	34.18 ± 1.77	31.94 ± 1.55	*37.33 ± 2.37*	36.22 ± 2.19	35.53 ± 2.10	33.49 ± 1.29	33.67 ± 2.04	32.37 ± 2.01
Youtobe-nisha-preprocessing	37.02 ± 1.82	33.14 ± 1.90	38.51 ± 1.25	35.49 ± 1.84	*40.45 ± 2.13*	39.22 ± 2.10	38.82 ± 2.22	37.49 ± 1.95	36.22 ± 2.37	31.22 ± 1.32
Z-alizadehsani	71.29 ± 1.47	79.87 ± 6.59	79.25 ± 8.32	78.26 ± 7.02	71.29 ± 1.47	71.29 ± 1.47	71.29 ± 1.47	*82.25 ± 7.86*	81.59 ± 7.34	70.66 ± 5.28
Zoo	91.18 ± 8.52	*96.18 ± 6.54*	92.18 ± 8.94	*96.18 ± 6.54*	94.18 ± 8.11	94.18 ± 8.11	94.18 ± 8.11	95.18 ± 6.65	95.18 ± 6.65	73.27 ± 10.54
Average acc (rank)	73.57 (5)	72.05 (8)	78.53 (1)	76.87 (4)	71.96 (9)	72.66 (7)	73.01 (6)	77.95 (2)	77.84 (3)	67.16 (10)
Average acc std (rank)	4.85 (4)	6.41 (10)	5.86 (6)	6.07 (7)	4.20 (1)	4.52 (2)	4.58 (3)	6.10 (8)	6.18 (9)	5.63 (5)

Table 7: The detailed experimental results on kappa statistic and standard deviation.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Abalone	0.38 ± 0.04	0.36 ± 0.03	0.41 ± 0.03	0.37 ± 0.04	0.50 ± 0.04	0.50 ± 0.04	0.50 ± 0.03	0.48 ± 0.05	0.48 ± 0.06	0.41 ± 0.03
Absenteeism-at-work	0.98 ± 0.03	0.71 ± 0.08	0.99 ± 0.01	0.67 ± 0.06	0.67 ± 0.07	0.67 ± 0.07	0.67 ± 0.07	0.95 ± 0.02	0.94 ± 0.05	0.90 ± 0.04
Acute-inflammation	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.58 ± 0.16
Acute-nephritis	1.00 ± 0.00	0.89 ± 0.15	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.84 ± 0.15
Adult	0.00 ± 0.00	0.34 ± 0.09	0.47 ± 0.10	0.41 ± 0.10	0.47 ± 0.13	0.49 ± 0.14	0.50 ± 0.14	0.44 ± 0.08	0.42 ± 0.08	0.26 ± 0.07
Aggregation	0.81 ± 0.07	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.01	1.00 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.47 ± 0.04
Algerianforest	0.78 ± 0.17	0.89 ± 0.10	0.91 ± 0.09	0.76 ± 0.14	0.06 ± 0.05	0.14 ± 0.09	0.14 ± 0.09	0.92 ± 0.08	0.93 ± 0.09	0.94 ± 0.10
Annealing	0.00 ± 0.00	0.50 ± 0.05	0.91 ± 0.05	0.74 ± 0.06	0.67 ± 0.10	0.76 ± 0.08	0.78 ± 0.06	0.78 ± 0.08	0.77 ± 0.08	0.44 ± 0.00
Arrhythmia	0.00 ± 0.00	0.43 ± 0.11	0.46 ± 0.10	0.24 ± 0.05	0.25 ± 0.05	0.35 ± 0.08	0.35 ± 0.08	0.46 ± 0.09	0.30 ± 0.12	0.21 ± 0.06
Au1-1000	0.00 ± 0.00	0.04 ± 0.06	0.35 ± 0.12	0.15 ± 0.11	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.04	0.17 ± 0.12	0.22 ± 0.11	0.00 ± 0.00
Au4-2500	0.30 ± 0.11	0.22 ± 0.08	0.51 ± 0.10	0.35 ± 0.08	0.29 ± 0.05	0.29 ± 0.05	0.29 ± 0.05	0.37 ± 0.08	0.37 ± 0.09	0.15 ± 0.08
Au6-1000	0.00 ± 0.02	0.02 ± 0.04	0.06 ± 0.05	-0.01 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.04	-0.01 ± 0.05	0.08 ± 0.03
Au6-250-drift-au6-cd1-500	0.02 ± 0.03	0.04 ± 0.05	0.05 ± 0.05	0.01 ± 0.05	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.00 ± 0.03	0.01 ± 0.04	0.08 ± 0.04
Au6-cd1-400	0.03 ± 0.09	0.08 ± 0.07	0.24 ± 0.08	0.01 ± 0.09	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.04 ± 0.08	0.03 ± 0.08	0.11 ± 0.08
Au7-300-drift-au7-cpd1-800	0.05 ± 0.05	0.15 ± 0.05	0.21 ± 0.06	0.14 ± 0.05	0.01 ± 0.02	0.01 ± 0.03	0.01 ± 0.03	0.15 ± 0.05	0.17 ± 0.06	0.03 ± 0.04
Au7-700	0.11 ± 0.10	0.14 ± 0.08	0.28 ± 0.06	0.03 ± 0.08	0.04 ± 0.04	0.03 ± 0.04	0.03 ± 0.04	0.11 ± 0.08	0.15 ± 0.05	0.17 ± 0.06
Au7-cpd1-500	0.03 ± 0.04	0.10 ± 0.08	0.37 ± 0.06	0.17 ± 0.08	-0.00 ± 0.01	0.01 ± 0.03	0.01 ± 0.03	0.18 ± 0.08	0.16 ± 0.07	0.12 ± 0.07
Audiology-std	0.03 ± 0.04	0.69 ± 0.12	0.72 ± 0.09	0.70 ± 0.14	0.55 ± 0.10	0.61 ± 0.13	0.65 ± 0.10	0.81 ± 0.08	0.80 ± 0.05	0.30 ± 0.02
Audit-risk	0.80 ± 0.09	0.88 ± 0.04	1.00 ± 0.01	0.95 ± 0.03	0.97 ± 0.03	0.97 ± 0.03	0.98 ± 0.03	0.93 ± 0.06	0.93 ± 0.06	1.00 ± 0.00
Autism-adolescent-data	0.55 ± 0.20	0.96 ± 0.09	1.00 ± 0.00	0.79 ± 0.15	0.92 ± 0.11	0.89 ± 0.11	0.92 ± 0.11	0.78 ± 0.22	0.78 ± 0.22	1.00 ± 0.00
Autism-adult-data	0.49 ± 0.13	0.93 ± 0.06	1.00 ± 0.00	0.87 ± 0.09	0.98 ± 0.03	0.99 ± 0.02	0.99 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Autism-child-data	0.83 ± 0.10	0.98 ± 0.03	1.00 ± 0.00	0.77 ± 0.07	0.99 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.02	0.99 ± 0.02	1.00 ± 0.00
Autos	0.62 ± 0.13	0.45 ± 0.16	0.78 ± 0.10	0.70 ± 0.13	0.03 ± 0.06	0.05 ± 0.07	0.05 ± 0.07	0.70 ± 0.08	0.70 ± 0.06	0.49 ± 0.13
Avila	0.96 ± 0.01	0.18 ± 0.02	0.92 ± 0.02	0.68 ± 0.03	0.56 ± 0.03	0.59 ± 0.03	0.61 ± 0.03	0.48 ± 0.03	0.49 ± 0.04	0.62 ± 0.03
Balance-scale	0.62 ± 0.05	0.82 ± 0.03	0.58 ± 0.06	0.67 ± 0.07	0.82 ± 0.03	0.87 ± 0.02	0.89 ± 0.02	0.84 ± 0.06	0.84 ± 0.06	0.19 ± 0.08
Balloons	0.50 ± 0.53	0.30 ± 0.48	0.40 ± 0.52	0.60 ± 0.52	0.50 ± 0.53	0.70 ± 0.48	0.70 ± 0.48	0.50 ± 0.53	0.50 ± 0.53	0.10 ± 0.57
Bank	0.00 ± 0.00	0.31 ± 0.06	0.36 ± 0.07	0.22 ± 0.06	0.27 ± 0.06	0.31 ± 0.05	0.34 ± 0.08	0.35 ± 0.08	0.37 ± 0.08	0.20 ± 0.07
Blood	0.05 ± 0.07	0.12 ± 0.08	0.34 ± 0.12	0.11 ± 0.11	0.20 ± 0.10	0.26 ± 0.05	0.26 ± 0.05	0.28 ± 0.09	0.30 ± 0.10	0.06 ± 0.08
Breast-cancer-wisc-diag	0.86 ± 0.08	0.85 ± 0.09	0.86 ± 0.08	0.91 ± 0.04	0.95 ± 0.05	0.95 ± 0.04	0.96 ± 0.02	0.92 ± 0.04	0.92 ± 0.04	0.77 ± 0.09
Breast-cancer-wisc-prog	0.00 ± 0.00	0.21 ± 0.29	0.29 ± 0.18	0.23 ± 0.29	0.20 ± 0.27	0.24 ± 0.34	0.24 ± 0.33	0.26 ± 0.26	0.27 ± 0.29	0.01 ± 0.21
Breast-cancer-wisc	0.91 ± 0.05	0.91 ± 0.04	0.86 ± 0.07	0.90 ± 0.07	0.94 ± 0.04	0.93 ± 0.04	0.93 ± 0.04	0.89 ± 0.08	0.90 ± 0.05	0.83 ± 0.11
Breast-cancer	0.01 ± 0.05	0.23 ± 0.18	0.21 ± 0.12	0.27 ± 0.18	0.23 ± 0.15	0.24 ± 0.16	0.22 ± 0.14	0.28 ± 0.18	0.18 ± 0.10	0.16 ± 0.13
Breast-tissue	0.61 ± 0.16	0.59 ± 0.18	0.59 ± 0.19	0.65 ± 0.17	0.49 ± 0.17	0.61 ± 0.20	0.63 ± 0.20	0.56 ± 0.11	0.55 ± 0.19	0.44 ± 0.12
Bupa	0.29 ± 0.11	0.19 ± 0.15	0.22 ± 0.18	0.21 ± 0.10	0.03 ± 0.05	0.03 ± 0.05	0.03 ± 0.05	0.30 ± 0.15	0.34 ± 0.14	0.05 ± 0.16
Caesarian	0.44 ± 0.31	0.31 ± 0.38	0.17 ± 0.35	0.25 ± 0.33	-0.02 ± 0.09	0.20 ± 0.25	0.16 ± 0.29	0.11 ± 0.41	0.11 ± 0.44	-0.00 ± 0.37
Car	0.00 ± 0.00	0.63 ± 0.09	0.95 ± 0.02	0.93 ± 0.03	0.94 ± 0.02	0.96 ± 0.02	0.96 ± 0.02	0.85 ± 0.03	0.86 ± 0.04	0.00 ± 0.00
Cardiotocography-10classes	0.58 ± 0.03	0.66 ± 0.03	0.80 ± 0.03	0.74 ± 0.03	0.77 ± 0.03	0.79 ± 0.03	0.80 ± 0.03	0.80 ± 0.04	0.78 ± 0.04	0.34 ± 0.03
Cardiotocography-3classes	0.00 ± 0.01	0.60 ± 0.06	0.81 ± 0.07	0.78 ± 0.05	0.75 ± 0.05	0.78 ± 0.06	0.78 ± 0.05	0.78 ± 0.03	0.78 ± 0.07	0.38 ± 0.12
Cervical-cancer	0.24 ± 0.30	0.76 ± 0.23	0.64 ± 0.33	0.71 ± 0.28	0.00 ± 0.00	0.12 ± 0.25	0.12 ± 0.25	0.82 ± 0.26	0.82 ± 0.26	0.37 ± 0.28
Chemicalcomposionofceramic	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.08	1.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Chess-krvk	0.15 ± 0.02	0.21 ± 0.02	0.46 ± 0.03	0.44 ± 0.04	0.39 ± 0.03	0.42 ± 0.03	0.43 ± 0.03	0.38 ± 0.03	0.36 ± 0.02	0.13 ± 0.02
Chess-krvkp	0.53 ± 0.03	0.71 ± 0.04	0.99 ± 0.01	0.91 ± 0.03	0.97 ± 0.02	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.32 ± 0.03
Congressional-voting	-0.00 ± 0.01	-0.02 ± 0.16	-0.03 ± 0.05	0.02 ± 0.13	0.04 ± 0.10	0.05 ± 0.12	0.04 ± 0.13	0.06 ± 0.17	0.04 ± 0.17	0.06 ± 0.07
Conn-bench-sonar-mines-rocks	0.53 ± 0.15	0.38 ± 0.18	0.42 ± 0.14	0.73 ± 0.14	0.68 ± 0.14	0.74 ± 0.16	0.77 ± 0.16	0.61 ± 0.22	0.64 ± 0.17	0.27 ± 0.21

Table 7: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Conn-bench-vowel-deterding	0.73 ± 0.03	0.62 ± 0.06	0.77 ± 0.07	0.99 ± 0.01	0.94 ± 0.02	0.97 ± 0.02	0.98 ± 0.02	0.83 ± 0.03	0.81 ± 0.04	0.29 ± 0.05
Connect-4	0.00 ± 0.00	0.13 ± 0.07	0.45 ± 0.05	0.25 ± 0.07	0.00 ± 0.01	0.11 ± 0.04	0.21 ± 0.06	0.29 ± 0.05	0.33 ± 0.05	0.00 ± 0.00
Connectionist	0.56 ± 0.18	0.36 ± 0.22	0.48 ± 0.15	0.74 ± 0.12	0.19 ± 0.17	0.37 ± 0.24	0.44 ± 0.25	0.67 ± 0.18	0.67 ± 0.17	0.25 ± 0.14
Contrac	0.12 ± 0.06	0.23 ± 0.08	0.27 ± 0.06	0.12 ± 0.05	0.28 ± 0.04	0.29 ± 0.05	0.28 ± 0.06	0.31 ± 0.05	0.32 ± 0.05	0.15 ± 0.04
Covid-19	0.60 ± 0.52	0.53 ± 0.50	0.63 ± 0.48	0.50 ± 0.55	0.40 ± 0.52	0.43 ± 0.50	0.53 ± 0.50	0.53 ± 0.50	0.53 ± 0.50	0.73 ± 0.44
Credit-approval	0.63 ± 0.09	0.52 ± 0.08	0.70 ± 0.08	0.64 ± 0.12	0.71 ± 0.11	0.70 ± 0.10	0.69 ± 0.09	0.70 ± 0.05	0.68 ± 0.08	0.71 ± 0.09
Crowdsource	0.29 ± 0.06	0.64 ± 0.05	0.76 ± 0.03	0.90 ± 0.02	0.27 ± 0.04	0.27 ± 0.04	0.27 ± 0.04	0.84 ± 0.04	0.84 ± 0.05	0.53 ± 0.05
Crx	0.66 ± 0.10	0.53 ± 0.07	0.72 ± 0.08	0.62 ± 0.10	0.03 ± 0.08	0.09 ± 0.11	0.12 ± 0.13	0.66 ± 0.09	0.67 ± 0.06	0.71 ± 0.09
Cryother	0.79 ± 0.28	0.66 ± 0.33	0.87 ± 0.16	0.80 ± 0.16	0.64 ± 0.26	0.66 ± 0.26	0.66 ± 0.26	0.75 ± 0.20	0.73 ± 0.18	0.62 ± 0.21
Cylinder-bands	0.16 ± 0.10	0.23 ± 0.15	0.41 ± 0.12	0.34 ± 0.16	0.46 ± 0.10	0.55 ± 0.11	0.60 ± 0.07	0.43 ± 0.11	0.45 ± 0.12	0.23 ± 0.10
Dbworld-bodies	0.07 ± 0.15	0.48 ± 0.34	0.61 ± 0.33	0.11 ± 0.22	0.00 ± 0.00	0.00 ± 0.00	0.22 ± 0.28	/	/	0.72 ± 0.27
Dbworld-bodies-stemmed	0.07 ± 0.15	0.51 ± 0.26	0.74 ± 0.27	0.24 ± 0.30	0.00 ± 0.00	0.11 ± 0.17	0.44 ± 0.29	/	/	0.51 ± 0.37
Dbworld-subjects	0.00 ± 0.00	0.79 ± 0.19	0.48 ± 0.21	0.61 ± 0.27	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.79 ± 0.27	0.76 ± 0.31	0.20 ± 0.26
Dbworld-subjects-stemmed	0.00 ± 0.00	0.73 ± 0.26	0.54 ± 0.33	0.68 ± 0.20	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.76 ± 0.24	0.73 ± 0.23	0.25 ± 0.19
Dermatology	0.81 ± 0.06	0.97 ± 0.03	0.96 ± 0.04	0.94 ± 0.05	0.97 ± 0.03	0.97 ± 0.03	0.97 ± 0.03	0.98 ± 0.02	0.97 ± 0.02	0.33 ± 0.04
Diabetes	0.29 ± 0.13	0.46 ± 0.13	0.41 ± 0.13	0.33 ± 0.12	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.45 ± 0.10	0.45 ± 0.12	0.32 ± 0.15
Diabetic	0.28 ± 0.09	0.17 ± 0.04	0.29 ± 0.09	0.23 ± 0.10	0.13 ± 0.10	0.16 ± 0.13	0.16 ± 0.12	0.44 ± 0.12	0.45 ± 0.12	0.06 ± 0.11
Divorce	0.95 ± 0.06	0.95 ± 0.06	0.90 ± 0.11	0.95 ± 0.06	0.95 ± 0.06	0.95 ± 0.06	0.95 ± 0.06	0.95 ± 0.06	0.95 ± 0.06	0.90 ± 0.11
Dota2train	0.00 ± 0.00	0.08 ± 0.07	0.08 ± 0.11	0.09 ± 0.08	0.00 ± 0.00	0.03 ± 0.08	0.03 ± 0.07	-0.00 ± 0.04	0.01 ± 0.04	0.08 ± 0.08
Dow-jones-index	0.30 ± 0.08	-0.01 ± 0.10	0.45 ± 0.08	0.09 ± 0.13	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.09 ± 0.11	0.09 ± 0.09	0.13 ± 0.06
Dry-bean-dataset	0.92 ± 0.02	0.88 ± 0.02	0.92 ± 0.01	0.93 ± 0.01	0.50 ± 0.04	0.50 ± 0.04	0.50 ± 0.04	0.91 ± 0.02	0.91 ± 0.02	0.61 ± 0.02
Early-stage-diabetes-data-upload	0.64 ± 0.16	0.73 ± 0.06	0.92 ± 0.06	0.96 ± 0.05	0.88 ± 0.09	0.89 ± 0.08	0.90 ± 0.08	0.92 ± 0.06	0.92 ± 0.07	0.65 ± 0.08
Echocardiogram	0.12 ± 0.21	0.53 ± 0.26	0.52 ± 0.22	0.43 ± 0.26	0.54 ± 0.22	0.53 ± 0.28	0.53 ± 0.28	0.52 ± 0.20	0.52 ± 0.25	0.63 ± 0.18
Ecoli	0.58 ± 0.07	0.82 ± 0.07	0.78 ± 0.10	0.73 ± 0.08	0.81 ± 0.06	0.81 ± 0.06	0.82 ± 0.05	0.81 ± 0.07	0.79 ± 0.05	0.53 ± 0.10
Eegeyesate	0.32 ± 0.03	0.01 ± 0.02	0.69 ± 0.02	0.67 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.06 ± 0.04	0.08 ± 0.05	0.22 ± 0.03
Electrical	0.62 ± 0.19	0.94 ± 0.11	1.00 ± 0.00	0.80 ± 0.16	0.58 ± 0.20	0.59 ± 0.17	0.62 ± 0.16	0.97 ± 0.04	0.97 ± 0.04	0.99 ± 0.02
Energy-y1	0.68 ± 0.02	0.70 ± 0.05	0.96 ± 0.02	0.62 ± 0.06	0.81 ± 0.02	0.83 ± 0.03	0.84 ± 0.05	0.82 ± 0.03	0.80 ± 0.03	0.74 ± 0.03
Energy-y2	0.84 ± 0.05	0.72 ± 0.05	0.84 ± 0.04	0.62 ± 0.05	0.84 ± 0.03	0.85 ± 0.02	0.86 ± 0.02	0.86 ± 0.04	0.86 ± 0.04	0.81 ± 0.05
Extentionofz-alizadehsani	0.00 ± 0.00	0.84 ± 0.09	0.99 ± 0.03	0.78 ± 0.11	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.95 ± 0.05	0.95 ± 0.05	0.71 ± 0.09
Fertility	0.00 ± 0.00	0.00 ± 0.00	-0.03 ± 0.06	0.14 ± 0.39	0.00 ± 0.00	0.10 ± 0.32	0.19 ± 0.43	0.43 ± 0.46	0.43 ± 0.47	0.00 ± 0.00
First-order	0.00 ± 0.00	0.03 ± 0.03	1.00 ± 0.00	0.97 ± 0.03	0.00 ± 0.00	0.66 ± 0.14	0.87 ± 0.07	0.99 ± 0.02	0.99 ± 0.02	0.18 ± 0.20
Flags	0.28 ± 0.14	0.34 ± 0.08	0.51 ± 0.10	0.31 ± 0.11	0.38 ± 0.11	0.41 ± 0.12	0.40 ± 0.15	0.34 ± 0.14	0.37 ± 0.13	0.45 ± 0.09
Foresttypes	0.92 ± 0.06	0.93 ± 0.06	0.93 ± 0.08	0.94 ± 0.05	0.01 ± 0.02	0.04 ± 0.04	0.04 ± 0.04	0.96 ± 0.05	0.96 ± 0.05	0.76 ± 0.11
Garments-worker-productivity	0.23 ± 0.03	0.07 ± 0.04	0.29 ± 0.08	0.02 ± 0.05	0.09 ± 0.04	0.10 ± 0.04	0.11 ± 0.04	0.20 ± 0.07	0.20 ± 0.05	0.08 ± 0.05
Gender-name-dataset	0.04 ± 0.04	0.00 ± 0.01	0.00 ± 0.00	0.16 ± 0.04	0.15 ± 0.03	0.16 ± 0.05	0.16 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.08 ± 0.03
Gesture-a1-raw	0.83 ± 0.03	0.61 ± 0.04	0.87 ± 0.03	0.95 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.87 ± 0.03	0.87 ± 0.02	0.95 ± 0.02
Gesture-a1-va3	0.48 ± 0.04	0.41 ± 0.05	0.50 ± 0.07	0.62 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.55 ± 0.03	0.55 ± 0.03	0.39 ± 0.03
Gesture-a2-raw	0.83 ± 0.03	0.56 ± 0.04	0.85 ± 0.04	0.92 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.85 ± 0.03	0.85 ± 0.03	0.92 ± 0.04
Gesture-a2-va3	0.40 ± 0.04	0.25 ± 0.08	0.38 ± 0.03	0.53 ± 0.06	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.43 ± 0.05	0.43 ± 0.06	0.30 ± 0.06
Gesture-a3-raw	0.87 ± 0.03	0.49 ± 0.05	0.90 ± 0.02	0.95 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.86 ± 0.02	0.85 ± 0.03	0.94 ± 0.02
Gesture-a3-va3	0.42 ± 0.04	0.31 ± 0.04	0.68 ± 0.04	0.86 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.51 ± 0.05	0.53 ± 0.04	0.30 ± 0.03
Gesture-b1-raw	0.88 ± 0.04	0.33 ± 0.07	0.90 ± 0.04	0.93 ± 0.04	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.83 ± 0.06	0.85 ± 0.03	0.88 ± 0.03
Gesture-b1-va3	0.43 ± 0.09	0.16 ± 0.08	0.59 ± 0.06	0.87 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.47 ± 0.05	0.49 ± 0.07	0.18 ± 0.06
Gesture-b3-raw	0.86 ± 0.04	0.45 ± 0.05	0.88 ± 0.04	0.94 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.85 ± 0.05	0.84 ± 0.05	0.94 ± 0.03

Table 7: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Gesture-b3-va3	0.25 ± 0.05	0.30 ± 0.05	0.44 ± 0.05	0.71 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.45 ± 0.07	0.42 ± 0.07	0.19 ± 0.08
Gesture-c1-raw	0.83 ± 0.04	0.54 ± 0.03	0.86 ± 0.03	0.93 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.84 ± 0.04	0.84 ± 0.05	0.92 ± 0.03
Gesture-c1-va3	0.39 ± 0.05	0.38 ± 0.05	0.46 ± 0.05	0.62 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.54 ± 0.04	0.52 ± 0.05	0.25 ± 0.06
Gesture-c3-raw	0.85 ± 0.03	0.49 ± 0.07	0.84 ± 0.05	0.91 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.82 ± 0.04	0.79 ± 0.03	0.93 ± 0.02
Gesture-c3-va3	0.31 ± 0.05	0.30 ± 0.03	0.42 ± 0.05	0.51 ± 0.06	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.45 ± 0.05	0.43 ± 0.07	0.25 ± 0.04
Glass	0.52 ± 0.16	0.36 ± 0.13	0.56 ± 0.11	0.60 ± 0.11	0.61 ± 0.10	0.60 ± 0.12	0.59 ± 0.12	0.54 ± 0.13	0.55 ± 0.08	0.38 ± 0.13
Go-track-tracks	0.67 ± 0.21	0.65 ± 0.09	0.75 ± 0.16	0.82 ± 0.17	0.44 ± 0.26	0.44 ± 0.24	0.44 ± 0.24	0.68 ± 0.21	0.69 ± 0.18	0.52 ± 0.26
Haberman-survival	0.03 ± 0.11	0.19 ± 0.16	0.17 ± 0.16	0.09 ± 0.16	0.11 ± 0.14	0.09 ± 0.10	0.09 ± 0.10	0.20 ± 0.19	0.23 ± 0.20	0.18 ± 0.15
Hayes-roth	0.47 ± 0.16	0.62 ± 0.20	0.70 ± 0.10	0.71 ± 0.10	0.79 ± 0.11	0.71 ± 0.10	0.70 ± 0.10	0.56 ± 0.20	0.54 ± 0.21	0.10 ± 0.11
Hcc-data	0.05 ± 0.13	0.32 ± 0.20	0.16 ± 0.16	0.24 ± 0.19	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.24 ± 0.32	0.23 ± 0.24	0.33 ± 0.24
Hcvdat	0.10 ± 0.11	0.66 ± 0.16	0.66 ± 0.09	0.52 ± 0.11	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.69 ± 0.14	0.67 ± 0.13	0.44 ± 0.13
Heart-cleveland	0.00 ± 0.01	0.31 ± 0.14	0.23 ± 0.10	0.29 ± 0.09	0.28 ± 0.14	0.32 ± 0.14	0.30 ± 0.14	0.27 ± 0.09	0.28 ± 0.10	0.10 ± 0.11
Heart-hungarian	0.52 ± 0.18	0.59 ± 0.17	0.52 ± 0.15	0.51 ± 0.18	0.59 ± 0.17	0.60 ± 0.15	0.62 ± 0.14	0.52 ± 0.18	0.51 ± 0.17	0.53 ± 0.21
Heart-switzerland	0.10 ± 0.12	0.18 ± 0.15	0.15 ± 0.17	0.06 ± 0.19	0.06 ± 0.10	0.10 ± 0.10	0.13 ± 0.11	0.19 ± 0.21	0.20 ± 0.17	-0.01 ± 0.18
Heart-va	0.05 ± 0.12	0.14 ± 0.13	0.12 ± 0.09	0.15 ± 0.10	0.13 ± 0.13	0.11 ± 0.11	0.09 ± 0.12	0.09 ± 0.13	0.07 ± 0.14	0.04 ± 0.13
Heart-failure-clinical-records-dataset	0.05 ± 0.07	0.41 ± 0.17	0.54 ± 0.14	0.15 ± 0.18	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.37 ± 0.12	0.47 ± 0.21	0.65 ± 0.17
Hepatitis	0.00 ± 0.00	0.51 ± 0.30	0.36 ± 0.33	0.41 ± 0.23	0.35 ± 0.29	0.42 ± 0.40	0.42 ± 0.40	0.40 ± 0.35	0.29 ± 0.34	-0.04 ± 0.19
Hill-valley	-0.00 ± 0.12	-0.02 ± 0.05	-0.00 ± 0.02	0.04 ± 0.07	0.04 ± 0.13	0.04 ± 0.15	0.06 ± 0.14	0.13 ± 0.15	0.07 ± 0.05	-0.05 ± 0.10
Hiv1625data	0.80 ± 0.06	0.82 ± 0.05	0.73 ± 0.08	0.78 ± 0.05	0.76 ± 0.07	0.81 ± 0.07	0.82 ± 0.07	0.87 ± 0.06	0.82 ± 0.07	0.32 ± 0.08
Hiv746data	0.85 ± 0.07	0.82 ± 0.07	0.65 ± 0.10	0.73 ± 0.07	0.79 ± 0.08	0.82 ± 0.06	0.82 ± 0.07	0.85 ± 0.05	0.85 ± 0.07	0.61 ± 0.09
Horse-colic	0.34 ± 0.09	0.43 ± 0.12	0.71 ± 0.12	0.50 ± 0.15	0.63 ± 0.11	0.66 ± 0.12	0.67 ± 0.13	0.56 ± 0.13	0.56 ± 0.18	0.59 ± 0.12
Htru	0.80 ± 0.04	0.71 ± 0.02	0.86 ± 0.03	0.83 ± 0.02	0.06 ± 0.03	0.15 ± 0.03	0.15 ± 0.03	0.87 ± 0.02	0.87 ± 0.02	0.85 ± 0.02
Hypothyroid	0.00 ± 0.00	0.60 ± 0.08	0.97 ± 0.02	0.36 ± 0.08	0.08 ± 0.06	0.16 ± 0.09	0.17 ± 0.09	0.56 ± 0.12	0.51 ± 0.13	0.76 ± 0.06
Ibeacon-rssi-labeled	0.23 ± 0.03	0.19 ± 0.02	0.25 ± 0.03	0.36 ± 0.04	0.33 ± 0.02	0.36 ± 0.03	0.37 ± 0.03	0.27 ± 0.04	0.27 ± 0.04	0.03 ± 0.02
Ilpd-indian-liver	0.02 ± 0.06	0.25 ± 0.06	0.01 ± 0.11	0.18 ± 0.11	0.00 ± 0.00	0.02 ± 0.08	0.07 ± 0.07	0.09 ± 0.12	0.10 ± 0.12	0.04 ± 0.10
Image-segmentation	0.86 ± 0.06	0.74 ± 0.06	0.87 ± 0.09	0.85 ± 0.05	0.85 ± 0.04	0.85 ± 0.05	0.84 ± 0.05	0.88 ± 0.06	0.88 ± 0.06	0.52 ± 0.10
Immunotherapy	0.00 ± 0.00	0.13 ± 0.33	0.37 ± 0.39	0.06 ± 0.45	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.37 ± 0.38	0.34 ± 0.28	0.49 ± 0.30
Impensdata	0.15 ± 0.09	0.58 ± 0.11	0.00 ± 0.00	0.42 ± 0.12	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.63 ± 0.09	0.51 ± 0.16	0.15 ± 0.07
In-vehicle-coupon-recommendation	0.01 ± 0.00	0.29 ± 0.03	0.39 ± 0.02	0.28 ± 0.03	0.36 ± 0.03	0.37 ± 0.03	0.38 ± 0.03	0.39 ± 0.02	0.41 ± 0.02	0.20 ± 0.05
Indian	0.02 ± 0.04	0.25 ± 0.06	0.18 ± 0.14	0.17 ± 0.11	0.05 ± 0.08	0.03 ± 0.07	0.03 ± 0.07	0.13 ± 0.14	0.13 ± 0.17	0.04 ± 0.10
Ionosphere	0.73 ± 0.08	0.64 ± 0.11	0.81 ± 0.07	0.68 ± 0.11	0.87 ± 0.05	0.89 ± 0.07	0.91 ± 0.07	0.78 ± 0.06	0.79 ± 0.08	0.58 ± 0.17
Iris	0.72 ± 0.17	0.90 ± 0.08	0.94 ± 0.08	0.93 ± 0.08	0.95 ± 0.05	0.94 ± 0.07	0.95 ± 0.05	0.96 ± 0.05	0.96 ± 0.05	0.88 ± 0.09
Jain	0.82 ± 0.13	0.86 ± 0.11	0.99 ± 0.04	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.87 ± 0.09	0.87 ± 0.09	0.86 ± 0.10
Jsbach-chorals-harmony	0.00 ± 0.00	0.84 ± 0.03	0.82 ± 0.03	0.88 ± 0.03	0.79 ± 0.04	0.84 ± 0.02	0.85 ± 0.04	0.88 ± 0.03	0.86 ± 0.05	0.79 ± 0.03
Knowledge	0.44 ± 0.10	0.82 ± 0.16	0.89 ± 0.09	0.64 ± 0.12	0.77 ± 0.14	0.84 ± 0.13	0.88 ± 0.11	0.89 ± 0.08	0.89 ± 0.08	0.75 ± 0.14
Lasvegastripadvisorreviews	-0.01 ± 0.04	0.03 ± 0.05	0.02 ± 0.06	0.01 ± 0.04	-0.02 ± 0.04	-0.04 ± 0.05	-0.01 ± 0.06	0.01 ± 0.08	0.02 ± 0.06	-0.00 ± 0.04
Leaf	0.57 ± 0.06	0.71 ± 0.09	0.59 ± 0.08	-0.00 ± 0.03	0.14 ± 0.04	0.21 ± 0.05	0.23 ± 0.06	0.49 ± 0.08	0.46 ± 0.08	0.21 ± 0.05
Led-display	0.67 ± 0.04	0.69 ± 0.03	0.68 ± 0.04	0.68 ± 0.04	0.69 ± 0.05	0.69 ± 0.04	0.68 ± 0.04	0.69 ± 0.04	0.69 ± 0.04	0.10 ± 0.01
Lenses	0.50 ± 0.53	0.50 ± 0.53	0.70 ± 0.42	0.75 ± 0.42	0.65 ± 0.47	0.70 ± 0.42	0.70 ± 0.42	0.50 ± 0.47	0.50 ± 0.47	0.53 ± 0.38
Letter	0.71 ± 0.01	0.63 ± 0.01	0.87 ± 0.01	0.96 ± 0.00	0.95 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	0.82 ± 0.01	0.81 ± 0.01	0.14 ± 0.00
Libras	0.65 ± 0.10	0.61 ± 0.11	0.68 ± 0.11	0.85 ± 0.05	0.80 ± 0.04	0.84 ± 0.05	0.86 ± 0.04	0.79 ± 0.06	0.80 ± 0.06	0.15 ± 0.06
Low-res-spect	0.40 ± 0.08	0.72 ± 0.05	0.74 ± 0.05	0.75 ± 0.06	0.84 ± 0.04	0.84 ± 0.04	0.86 ± 0.04	0.88 ± 0.05	0.87 ± 0.04	0.60 ± 0.08
Lung-cancer	0.32 ± 0.34	0.48 ± 0.34	0.09 ± 0.42	0.27 ± 0.56	0.25 ± 0.35	0.36 ± 0.34	0.26 ± 0.40	0.18 ± 0.46	0.23 ± 0.39	0.19 ± 0.30
Lymphography	0.51 ± 0.16	0.65 ± 0.17	0.52 ± 0.16	0.53 ± 0.25	0.72 ± 0.18	0.71 ± 0.19	0.67 ± 0.17	0.62 ± 0.19	0.60 ± 0.19	0.53 ± 0.17

Table 7: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Magic	0.42 ± 0.04	0.35 ± 0.06	0.61 ± 0.05	0.55 ± 0.05	0.65 ± 0.05	0.65 ± 0.04	0.65 ± 0.03	0.65 ± 0.05	0.64 ± 0.04	0.37 ± 0.10
Mammographic	0.57 ± 0.07	0.57 ± 0.07	0.64 ± 0.04	0.50 ± 0.10	0.65 ± 0.04	0.64 ± 0.05	0.65 ± 0.05	0.63 ± 0.06	0.62 ± 0.06	0.63 ± 0.03
Miniboone	0.30 ± 0.09	0.02 ± 0.01	0.64 ± 0.06	0.58 ± 0.05	0.54 ± 0.08	0.59 ± 0.07	0.61 ± 0.06	0.56 ± 0.05	0.56 ± 0.05	0.49 ± 0.06
Molec-biol-promoter	0.77 ± 0.33	0.77 ± 0.22	0.49 ± 0.20	0.45 ± 0.24	0.65 ± 0.27	0.69 ± 0.24	0.69 ± 0.24	0.56 ± 0.22	0.54 ± 0.21	0.39 ± 0.22
Molec-biol-splice	0.00 ± 0.00	0.88 ± 0.03	0.88 ± 0.03	0.43 ± 0.03	0.79 ± 0.03	0.79 ± 0.03	0.79 ± 0.02	0.75 ± 0.04	0.75 ± 0.03	0.41 ± 0.03
Monks-1	0.84 ± 0.19	0.43 ± 0.23	0.93 ± 0.14	0.41 ± 0.20	0.61 ± 0.24	0.66 ± 0.22	0.66 ± 0.19	0.77 ± 0.30	0.79 ± 0.34	0.46 ± 0.20
Monks-2	0.00 ± 0.00	-0.09 ± 0.21	0.50 ± 0.24	0.15 ± 0.24	-0.00 ± 0.14	0.12 ± 0.27	0.21 ± 0.26	0.43 ± 0.26	0.45 ± 0.17	-0.04 ± 0.07
Monks-3	0.87 ± 0.17	0.77 ± 0.12	0.87 ± 0.17	0.49 ± 0.33	0.80 ± 0.15	0.79 ± 0.16	0.79 ± 0.16	0.68 ± 0.22	0.78 ± 0.14	0.56 ± 0.08
Mushroom	0.81 ± 0.02	0.78 ± 0.02	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.01
Musk-1	0.67 ± 0.12	0.48 ± 0.13	0.70 ± 0.13	0.70 ± 0.09	0.81 ± 0.09	0.85 ± 0.06	0.87 ± 0.06	0.88 ± 0.05	0.87 ± 0.06	0.22 ± 0.18
Musk-2	0.33 ± 0.07	0.50 ± 0.06	0.83 ± 0.06	0.86 ± 0.03	0.84 ± 0.05	0.90 ± 0.04	0.92 ± 0.03	0.96 ± 0.03	0.95 ± 0.03	0.54 ± 0.07
Newdiagnosis	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.55 ± 0.14
Nursery	0.82 ± 0.02	0.85 ± 0.01	0.99 ± 0.00	0.84 ± 0.01	0.97 ± 0.00	0.98 ± 0.00	0.98 ± 0.00	0.94 ± 0.01	0.92 ± 0.02	0.57 ± 0.02
Obesitydataset-raw-and-data-sinthetic	0.80 ± 0.03	0.62 ± 0.03	0.93 ± 0.02	0.79 ± 0.03	0.88 ± 0.03	0.90 ± 0.03	0.91 ± 0.03	0.93 ± 0.03	0.93 ± 0.02	0.62 ± 0.04
Obs-network-dataset-2-aug27	0.94 ± 0.03	0.60 ± 0.07	0.99 ± 0.02	0.99 ± 0.01	0.97 ± 0.01	0.97 ± 0.01	0.97 ± 0.01	0.95 ± 0.02	0.97 ± 0.02	0.86 ± 0.03
Occupancy-data	0.92 ± 0.02	0.91 ± 0.02	0.97 ± 0.02	0.98 ± 0.01	0.54 ± 0.06	0.57 ± 0.06	0.57 ± 0.06	0.96 ± 0.02	0.96 ± 0.02	0.97 ± 0.02
Occupancy-data2	0.95 ± 0.01	0.89 ± 0.01	0.98 ± 0.00	0.98 ± 0.01	0.30 ± 0.03	0.42 ± 0.03	0.42 ± 0.03	0.98 ± 0.01	0.98 ± 0.00	0.98 ± 0.00
Occupancy-data3	0.95 ± 0.02	0.93 ± 0.02	0.99 ± 0.00	0.98 ± 0.01	0.28 ± 0.03	0.37 ± 0.03	0.37 ± 0.03	0.96 ± 0.01	0.96 ± 0.01	0.98 ± 0.01
Old	0.76 ± 0.04	0.88 ± 0.02	0.90 ± 0.03	0.95 ± 0.02	0.90 ± 0.02	0.90 ± 0.03	0.91 ± 0.02	0.94 ± 0.02	0.94 ± 0.02	0.77 ± 0.03
Online-shoppers-intention	0.00 ± 0.00	0.42 ± 0.03	0.57 ± 0.02	0.26 ± 0.03	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.01	0.56 ± 0.03	0.55 ± 0.06	0.52 ± 0.04
Oocytes-merluccius-nucleus-4d	0.24 ± 0.07	0.20 ± 0.08	0.42 ± 0.08	0.39 ± 0.08	0.40 ± 0.09	0.49 ± 0.07	0.52 ± 0.08	0.58 ± 0.04	0.56 ± 0.10	0.16 ± 0.14
Oocytes-merluccius-states-2f	0.74 ± 0.08	0.70 ± 0.07	0.80 ± 0.07	0.81 ± 0.06	0.82 ± 0.05	0.83 ± 0.04	0.84 ± 0.04	0.82 ± 0.03	0.81 ± 0.06	0.62 ± 0.10
Oocytes-trisopterus-nucleus-2f	0.23 ± 0.11	0.13 ± 0.10	0.43 ± 0.13	0.48 ± 0.11	0.60 ± 0.08	0.66 ± 0.07	0.65 ± 0.06	0.65 ± 0.08	0.63 ± 0.09	0.16 ± 0.13
Oocytes-trisopterus-states-5b	0.65 ± 0.10	0.51 ± 0.09	0.77 ± 0.05	0.82 ± 0.08	0.83 ± 0.07	0.84 ± 0.06	0.85 ± 0.05	0.87 ± 0.05	0.87 ± 0.06	0.58 ± 0.06
Optdigits	0.88 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.98 ± 0.01	0.70 ± 0.02	0.72 ± 0.02	0.72 ± 0.02	0.98 ± 0.01	0.98 ± 0.01	0.19 ± 0.01
Optical	0.87 ± 0.02	0.91 ± 0.01	0.89 ± 0.02	0.98 ± 0.01	0.98 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.20 ± 0.01
Ozone	0.00 ± 0.00	0.09 ± 0.03	0.20 ± 0.13	0.16 ± 0.12	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.30 ± 0.18	0.32 ± 0.16	-0.00 ± 0.01
Page-blocks	0.45 ± 0.09	0.55 ± 0.05	0.83 ± 0.02	0.78 ± 0.03	0.77 ± 0.05	0.79 ± 0.05	0.80 ± 0.05	0.78 ± 0.04	0.78 ± 0.03	0.64 ± 0.03
Parkingbirmingham	0.80 ± 0.02	0.75 ± 0.02	1.00 ± 0.00	0.97 ± 0.01	0.57 ± 0.04	0.60 ± 0.04	0.60 ± 0.04	0.86 ± 0.02	0.87 ± 0.02	1.00 ± 0.00
Parkinsons	0.45 ± 0.26	0.40 ± 0.19	0.47 ± 0.16	0.91 ± 0.07	0.59 ± 0.23	0.67 ± 0.21	0.73 ± 0.19	0.76 ± 0.19	0.72 ± 0.23	0.57 ± 0.25
Pasture	0.59 ± 0.34	0.62 ± 0.35	0.66 ± 0.34	0.58 ± 0.41	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.64 ± 0.40	0.60 ± 0.42	0.47 ± 0.36
Pbc	0.43 ± 0.12	0.51 ± 0.16	0.50 ± 0.19	0.21 ± 0.18	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.39 ± 0.11	0.43 ± 0.08	0.39 ± 0.19
Pen	0.86 ± 0.01	0.84 ± 0.01	0.96 ± 0.01	0.99 ± 0.00	0.03 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.94 ± 0.01	0.94 ± 0.00	0.32 ± 0.02
Pendigits	0.85 ± 0.01	0.87 ± 0.02	0.96 ± 0.01	0.99 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	0.95 ± 0.01	0.94 ± 0.01	0.33 ± 0.02
Pharynx	0.00 ± 0.00	-0.06 ± 0.19	0.00 ± 0.00	-0.08 ± 0.24	0.07 ± 0.22	0.09 ± 0.29	0.04 ± 0.27	0.01 ± 0.16	0.02 ± 0.06	-0.01 ± 0.08
Phishingwebsites	0.69 ± 0.02	0.86 ± 0.01	0.92 ± 0.01	0.94 ± 0.01	0.89 ± 0.01	0.90 ± 0.01	0.90 ± 0.01	0.93 ± 0.01	0.94 ± 0.01	0.77 ± 0.01
Pima	0.28 ± 0.12	0.46 ± 0.13	0.41 ± 0.13	0.33 ± 0.12	0.44 ± 0.09	0.43 ± 0.09	0.43 ± 0.08	0.45 ± 0.10	0.45 ± 0.12	0.32 ± 0.15
Pittsburg-bridges-rel-l	0.39 ± 0.19	0.42 ± 0.24	0.33 ± 0.27	0.51 ± 0.29	0.37 ± 0.26	0.40 ± 0.26	0.51 ± 0.20	0.35 ± 0.26	0.41 ± 0.21	0.46 ± 0.21
Pittsburg-bridges-span	0.24 ± 0.18	0.44 ± 0.25	0.32 ± 0.25	0.29 ± 0.28	0.39 ± 0.25	0.38 ± 0.28	0.37 ± 0.28	0.42 ± 0.18	0.46 ± 0.22	0.12 ± 0.23
Pittsburg-bridges-t-or-d	0.00 ± 0.00	0.19 ± 0.40	0.09 ± 0.34	0.30 ± 0.43	0.09 ± 0.32	0.24 ± 0.45	0.31 ± 0.45	0.30 ± 0.53	0.29 ± 0.54	0.38 ± 0.43
Pittsburg-bridges-type	0.32 ± 0.17	0.41 ± 0.16	0.48 ± 0.13	0.44 ± 0.16	0.34 ± 0.17	0.47 ± 0.24	0.49 ± 0.19	0.47 ± 0.22	0.46 ± 0.21	0.36 ± 0.09
Pittsburg-bridgesmaterial	0.50 ± 0.28	0.59 ± 0.17	0.65 ± 0.23	0.61 ± 0.25	0.62 ± 0.15	0.60 ± 0.13	0.59 ± 0.21	0.60 ± 0.18	0.52 ± 0.16	0.62 ± 0.14
Planning	0.03 ± 0.08	-0.01 ± 0.15	0.00 ± 0.00	0.12 ± 0.20	0.00 ± 0.00	0.01 ± 0.15	-0.05 ± 0.14	0.00 ± 0.18	-0.07 ± 0.20	-0.08 ± 0.18
Plant-margin	0.63 ± 0.04	0.85 ± 0.03	0.48 ± 0.05	0.74 ± 0.03	0.83 ± 0.02	0.85 ± 0.03	0.85 ± 0.03	0.83 ± 0.02	0.83 ± 0.03	0.07 ± 0.02

Table 7: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Plant-shape	0.49 ± 0.04	0.53 ± 0.03	0.46 ± 0.03	0.64 ± 0.03	0.48 ± 0.02	0.56 ± 0.03	0.60 ± 0.03	0.64 ± 0.04	0.65 ± 0.03	0.08 ± 0.02
Plant-texture	0.62 ± 0.03	0.74 ± 0.03	0.51 ± 0.04	0.80 ± 0.03	0.83 ± 0.03	0.85 ± 0.02	0.86 ± 0.02	0.84 ± 0.02	0.83 ± 0.02	0.03 ± 0.01
Poker-hand-training-true	0.00 ± 0.00	0.00 ± 0.00	0.20 ± 0.03	0.06 ± 0.01	0.22 ± 0.02	0.21 ± 0.02	0.20 ± 0.02	0.11 ± 0.02	0.08 ± 0.03	0.00 ± 0.00
Post-operative	0.00 ± 0.00	-0.03 ± 0.14	0.00 ± 0.00	-0.06 ± 0.24	0.00 ± 0.00	-0.02 ± 0.10	-0.02 ± 0.16	-0.10 ± 0.22	-0.09 ± 0.27	-0.03 ± 0.07
Primary-tumor	0.01 ± 0.02	0.36 ± 0.11	0.35 ± 0.10	0.27 ± 0.09	0.37 ± 0.13	0.36 ± 0.12	0.35 ± 0.14	0.33 ± 0.09	0.33 ± 0.10	0.08 ± 0.03
Qsarbioconcentration	-0.00 ± 0.01	-0.01 ± 0.04	-0.03 ± 0.06	0.01 ± 0.15	-0.00 ± 0.01	-0.02 ± 0.05	-0.05 ± 0.06	-0.03 ± 0.08	0.00 ± 0.07	-0.01 ± 0.08
Qsarbiodegradation	0.15 ± 0.07	0.52 ± 0.06	0.63 ± 0.08	0.64 ± 0.05	0.68 ± 0.06	0.69 ± 0.05	0.67 ± 0.06	0.70 ± 0.05	0.71 ± 0.06	0.46 ± 0.09
Qualitative-bankruptcy	0.98 ± 0.05	0.98 ± 0.05	0.96 ± 0.06	1.00 ± 0.00	0.97 ± 0.06	0.97 ± 0.06	0.99 ± 0.03	0.98 ± 0.05	0.98 ± 0.05	0.97 ± 0.06
Ringnorm	0.58 ± 0.03	0.97 ± 0.01	0.83 ± 0.01	0.50 ± 0.03	0.97 ± 0.01	0.97 ± 0.01	0.97 ± 0.01	0.84 ± 0.03	0.83 ± 0.02	0.29 ± 0.02
Risk-factors-cervical-cancer	0.00 ± 0.00	0.41 ± 0.09	0.59 ± 0.19	0.49 ± 0.17	0.00 ± 0.00	0.02 ± 0.09	0.23 ± 0.21	0.53 ± 0.19	0.50 ± 0.22	0.73 ± 0.13
Robotnavigation	0.88 ± 0.02	0.36 ± 0.03	0.99 ± 0.00	0.82 ± 0.02	0.85 ± 0.02	0.87 ± 0.01	0.88 ± 0.01	0.82 ± 0.03	0.82 ± 0.04	0.61 ± 0.03
Sapfile	0.15 ± 0.18	0.30 ± 0.15	0.16 ± 0.21	0.09 ± 0.11	0.14 ± 0.14	0.17 ± 0.11	0.19 ± 0.13	0.19 ± 0.14	0.17 ± 0.19	0.05 ± 0.20
Sat	0.81 ± 0.01	0.75 ± 0.02	0.83 ± 0.01	0.88 ± 0.01	0.03 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.87 ± 0.02	0.86 ± 0.01	0.50 ± 0.02
Satelite	0.81 ± 0.01	0.75 ± 0.01	0.83 ± 0.01	0.88 ± 0.01	0.03 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.87 ± 0.01	0.87 ± 0.01	0.51 ± 0.02
Scadi	0.34 ± 0.11	0.76 ± 0.13	0.72 ± 0.19	0.72 ± 0.19	0.66 ± 0.12	0.70 ± 0.19	0.74 ± 0.21	0.72 ± 0.14	0.72 ± 0.14	0.41 ± 0.12
Schillingdata	0.02 ± 0.02	0.71 ± 0.04	0.00 ± 0.00	0.43 ± 0.06	0.00 ± 0.00	0.27 ± 0.09	0.53 ± 0.05	0.69 ± 0.06	0.60 ± 0.10	0.01 ± 0.02
Seeds	0.80 ± 0.10	0.87 ± 0.07	0.88 ± 0.09	0.91 ± 0.03	0.91 ± 0.06	0.91 ± 0.06	0.91 ± 0.06	0.93 ± 0.05	0.94 ± 0.05	0.74 ± 0.15
Segment	0.89 ± 0.03	0.77 ± 0.02	0.96 ± 0.01	0.97 ± 0.01	0.60 ± 0.03	0.62 ± 0.03	0.61 ± 0.03	0.95 ± 0.01	0.96 ± 0.01	0.58 ± 0.03
Seismic-bumps	0.00 ± 0.00	0.22 ± 0.09	-0.00 ± 0.00	0.11 ± 0.08	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.06 ± 0.09	0.07 ± 0.06	0.00 ± 0.03
Semeion	0.76 ± 0.04	0.84 ± 0.03	0.73 ± 0.03	0.91 ± 0.02	0.95 ± 0.01	0.96 ± 0.01	0.96 ± 0.01	0.92 ± 0.02	0.92 ± 0.02	0.11 ± 0.00
Setapprocesst1	0.06 ± 0.19	0.16 ± 0.41	0.17 ± 0.45	0.15 ± 0.24	-0.08 ± 0.13	0.07 ± 0.31	0.03 ± 0.24	0.15 ± 0.47	0.13 ± 0.49	0.38 ± 0.47
Setapprocesst10	0.00 ± 0.00	0.01 ± 0.28	0.05 ± 0.32	0.09 ± 0.40	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.03 ± 0.50	-0.01 ± 0.41	0.00 ± 0.20
Setapprocesst11	0.00 ± 0.00	0.08 ± 0.37	0.24 ± 0.44	0.08 ± 0.33	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.20 ± 0.29	0.29 ± 0.33	-0.03 ± 0.21
Setapprocesst2	0.06 ± 0.19	0.25 ± 0.36	0.34 ± 0.50	0.14 ± 0.48	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.18 ± 0.40	0.24 ± 0.41	0.53 ± 0.44
Setapprocesst3	0.00 ± 0.00	0.22 ± 0.30	0.12 ± 0.35	0.18 ± 0.38	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.23 ± 0.32	0.18 ± 0.28	-0.02 ± 0.35
Setapprocesst4	0.00 ± 0.00	0.01 ± 0.46	-0.12 ± 0.30	-0.02 ± 0.41	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	-0.04 ± 0.43	-0.07 ± 0.39	0.23 ± 0.34
Setapprocesst5	0.10 ± 0.21	0.01 ± 0.30	-0.11 ± 0.32	0.06 ± 0.34	-0.02 ± 0.07	-0.05 ± 0.10	-0.05 ± 0.10	0.06 ± 0.22	0.11 ± 0.26	0.25 ± 0.46
Setapprocesst6	0.00 ± 0.00	0.24 ± 0.36	0.14 ± 0.37	0.09 ± 0.35	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.11 ± 0.42	0.10 ± 0.37	0.05 ± 0.40
Setapprocesst7	0.06 ± 0.19	0.40 ± 0.38	0.14 ± 0.45	0.09 ± 0.36	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.32 ± 0.46	0.33 ± 0.43	-0.04 ± 0.34
Setapprocesst8	0.00 ± 0.00	0.35 ± 0.25	0.37 ± 0.32	0.08 ± 0.39	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.23 ± 0.39	0.21 ± 0.42	-0.15 ± 0.20
Setapprocesst9	0.00 ± 0.00	0.27 ± 0.34	0.35 ± 0.36	-0.04 ± 0.23	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.32 ± 0.42	0.27 ± 0.36	-0.08 ± 0.30
Shillbiddingdataset	0.25 ± 0.06	0.85 ± 0.02	0.98 ± 0.01	0.96 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.98 ± 0.01	0.99 ± 0.01	0.87 ± 0.02
Shuttle-landing-control	0.90 ± 0.32	0.70 ± 0.48	0.90 ± 0.32	0.90 ± 0.32	0.90 ± 0.32	0.90 ± 0.32	0.90 ± 0.32	0.80 ± 0.42	0.70 ± 0.48	0.80 ± 0.42
Somervillehappinesssurvey2015	0.12 ± 0.26	0.18 ± 0.20	0.28 ± 0.34	0.16 ± 0.25	0.19 ± 0.22	0.16 ± 0.20	0.16 ± 0.23	0.17 ± 0.19	0.18 ± 0.23	0.31 ± 0.32
Sonar	0.52 ± 0.15	0.37 ± 0.18	0.42 ± 0.14	0.73 ± 0.14	0.29 ± 0.11	0.39 ± 0.18	0.46 ± 0.19	0.64 ± 0.21	0.63 ± 0.19	0.25 ± 0.22
Soybean	0.65 ± 0.07	0.91 ± 0.06	0.86 ± 0.06	0.86 ± 0.05	0.89 ± 0.07	0.91 ± 0.05	0.91 ± 0.05	0.89 ± 0.08	0.90 ± 0.07	0.22 ± 0.04
Spambase	0.48 ± 0.03	0.59 ± 0.03	0.85 ± 0.03	0.81 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.81 ± 0.03	0.78 ± 0.13	0.54 ± 0.04
Speaker-accent	0.04 ± 0.04	0.47 ± 0.12	0.57 ± 0.13	0.73 ± 0.12	0.22 ± 0.08	0.32 ± 0.10	0.32 ± 0.10	0.71 ± 0.11	0.64 ± 0.12	0.13 ± 0.08
Spect	0.14 ± 0.23	0.36 ± 0.42	0.19 ± 0.40	-0.08 ± 0.36	0.04 ± 0.28	0.06 ± 0.34	0.05 ± 0.31	-0.05 ± 0.44	-0.03 ± 0.38	0.19 ± 0.28
Spectf	0.38 ± 0.24	0.50 ± 0.24	0.47 ± 0.25	0.35 ± 0.38	0.35 ± 0.13	0.35 ± 0.13	0.53 ± 0.28	0.42 ± 0.29	0.45 ± 0.31	0.43 ± 0.17
Statlog-australian-credit	-0.00 ± 0.01	0.09 ± 0.08	0.15 ± 0.15	0.00 ± 0.06	0.02 ± 0.04	-0.00 ± 0.12	0.02 ± 0.12	0.06 ± 0.13	0.04 ± 0.08	0.09 ± 0.15
Statlog-german-credit	0.00 ± 0.00	0.39 ± 0.14	0.35 ± 0.12	0.23 ± 0.10	0.36 ± 0.11	0.41 ± 0.09	0.42 ± 0.11	0.30 ± 0.07	0.32 ± 0.08	0.12 ± 0.06
Statlog-heart	0.66 ± 0.16	0.66 ± 0.15	0.52 ± 0.20	0.50 ± 0.17	0.65 ± 0.10	0.60 ± 0.16	0.59 ± 0.19	0.56 ± 0.14	0.60 ± 0.17	0.41 ± 0.12
Statlog-image	0.89 ± 0.03	0.77 ± 0.02	0.96 ± 0.01	0.97 ± 0.01	0.93 ± 0.01	0.94 ± 0.01	0.95 ± 0.01	0.95 ± 0.01	0.96 ± 0.01	0.58 ± 0.03

Table 7: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Statlog-landsat	0.82 ± 0.02	0.75 ± 0.03	0.83 ± 0.02	0.88 ± 0.02	0.87 ± 0.02	0.88 ± 0.02	0.88 ± 0.02	0.86 ± 0.02	0.86 ± 0.02	0.51 ± 0.02
Statlog-shuttle	0.38 ± 0.02	0.75 ± 0.03	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.86 ± 0.01
Statlog-vehicle	0.55 ± 0.04	0.27 ± 0.04	0.63 ± 0.08	0.60 ± 0.06	0.69 ± 0.05	0.74 ± 0.03	0.74 ± 0.03	0.76 ± 0.04	0.74 ± 0.06	0.35 ± 0.07
Steel-plates	0.46 ± 0.04	0.52 ± 0.05	0.69 ± 0.04	0.64 ± 0.03	0.68 ± 0.04	0.70 ± 0.03	0.70 ± 0.04	0.65 ± 0.04	0.65 ± 0.04	0.30 ± 0.05
Synthetic-control	0.87 ± 0.03	0.94 ± 0.04	0.90 ± 0.04	0.96 ± 0.03	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.48 ± 0.05
Teaching	0.25 ± 0.17	0.31 ± 0.13	0.39 ± 0.17	0.43 ± 0.20	0.34 ± 0.22	0.42 ± 0.19	0.43 ± 0.18	0.31 ± 0.17	0.28 ± 0.14	0.20 ± 0.16
Thoraricsurgery	0.00 ± 0.00	0.05 ± 0.15	0.00 ± 0.04	0.04 ± 0.17	0.00 ± 0.00	-0.01 ± 0.02	-0.01 ± 0.02	0.11 ± 0.13	0.15 ± 0.20	-0.01 ± 0.04
Thyroid	0.61 ± 0.21	0.92 ± 0.09	0.83 ± 0.14	0.94 ± 0.05	0.27 ± 0.19	0.34 ± 0.17	0.34 ± 0.17	0.93 ± 0.07	0.93 ± 0.07	0.79 ± 0.20
Thyroid-train	0.00 ± 0.00	0.63 ± 0.10	0.98 ± 0.02	0.39 ± 0.10	0.51 ± 0.10	0.60 ± 0.08	0.67 ± 0.08	0.72 ± 0.10	0.68 ± 0.09	0.78 ± 0.05
Tic-tac-toe	0.00 ± 0.00	0.27 ± 0.08	0.86 ± 0.05	1.00 ± 0.00	0.97 ± 0.03	0.98 ± 0.02	0.99 ± 0.02	0.95 ± 0.02	0.95 ± 0.03	0.34 ± 0.05
Titanic	0.40 ± 0.07	0.42 ± 0.06	0.43 ± 0.05	0.43 ± 0.05	0.43 ± 0.06	0.43 ± 0.06	0.43 ± 0.06	0.44 ± 0.06	0.42 ± 0.06	0.44 ± 0.05
Trains	0.70 ± 0.48	0.60 ± 0.52	0.70 ± 0.48	0.60 ± 0.52	0.60 ± 0.52	0.70 ± 0.48	0.70 ± 0.48	0.80 ± 0.42	0.80 ± 0.42	0.00 ± 0.00
Transfusion	0.05 ± 0.07	0.16 ± 0.10	0.34 ± 0.12	0.12 ± 0.12	0.08 ± 0.11	0.06 ± 0.09	0.04 ± 0.10	0.28 ± 0.09	0.30 ± 0.10	0.06 ± 0.08
Trial	0.87 ± 0.07	0.80 ± 0.07	1.00 ± 0.00	0.99 ± 0.02	0.98 ± 0.02	0.99 ± 0.01	0.99 ± 0.01	0.98 ± 0.02	0.99 ± 0.02	1.00 ± 0.00
Turkiye-student-evaluation	0.00 ± 0.00	0.75 ± 0.03	1.00 ± 0.00	0.82 ± 0.03	0.91 ± 0.01	0.93 ± 0.01	0.93 ± 0.01	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Unbalanced	0.00 ± 0.00	0.07 ± 0.13	0.00 ± 0.00	0.16 ± 0.36	0.00 ± 0.00	-0.00 ± 0.00	-0.01 ± 0.01	-0.00 ± 0.01	0.04 ± 0.16	0.00 ± 0.00
Urbanlandcover	0.77 ± 0.12	0.75 ± 0.11	0.76 ± 0.09	0.73 ± 0.13	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.74 ± 0.14	0.74 ± 0.14	0.42 ± 0.16
Userknowledgemodeling	0.48 ± 0.11	0.85 ± 0.08	0.92 ± 0.05	0.73 ± 0.14	0.67 ± 0.10	0.79 ± 0.06	0.79 ± 0.06	0.88 ± 0.05	0.90 ± 0.07	0.79 ± 0.11
Vehicle	0.59 ± 0.06	0.27 ± 0.06	0.63 ± 0.08	0.60 ± 0.06	0.07 ± 0.03	0.08 ± 0.04	0.08 ± 0.04	*0.76 ± 0.05*	0.75 ± 0.06	0.36 ± 0.07
Vertebral-column-2classes	0.34 ± 0.23	0.55 ± 0.13	0.57 ± 0.16	0.60 ± 0.11	*0.66 ± 0.13*	0.64 ± 0.15	0.65 ± 0.14	0.63 ± 0.17	0.62 ± 0.16	0.43 ± 0.15
Vertebral-column-3classes	0.50 ± 0.18	0.73 ± 0.09	0.72 ± 0.09	0.66 ± 0.08	0.74 ± 0.09	0.76 ± 0.11	0.75 ± 0.10	*0.77 ± 0.07*	0.76 ± 0.09	0.59 ± 0.11
Veteran	0.03 ± 0.10	0.22 ± 0.21	*0.36 ± 0.17*	0.07 ± 0.21	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.06 ± 0.34	0.19 ± 0.35	0.35 ± 0.18
Vowel	0.59 ± 0.04	0.60 ± 0.05	0.80 ± 0.05	*0.99 ± 0.01*	0.84 ± 0.03	0.90 ± 0.02	0.92 ± 0.02	0.92 ± 0.04	0.93 ± 0.03	0.25 ± 0.05
Wall-following	0.88 ± 0.02	0.36 ± 0.02	*0.99 ± 0.00*	0.82 ± 0.02	0.83 ± 0.02	0.86 ± 0.02	0.87 ± 0.02	0.81 ± 0.02	0.82 ± 0.03	0.61 ± 0.02
Waveform-noise	0.51 ± 0.03	0.70 ± 0.03	0.63 ± 0.02	0.60 ± 0.02	*0.79 ± 0.02*	0.78 ± 0.02	0.77 ± 0.02	0.75 ± 0.02	0.75 ± 0.02	0.31 ± 0.04
Waveform	0.56 ± 0.02	0.72 ± 0.02	0.64 ± 0.02	0.65 ± 0.03	*0.80 ± 0.02*	0.78 ± 0.02	0.78 ± 0.02	0.76 ± 0.02	0.75 ± 0.02	0.29 ± 0.02
Wbc	0.93 ± 0.04	*0.95 ± 0.03*	0.84 ± 0.05	0.91 ± 0.06	*0.95 ± 0.03*	*0.95 ± 0.03*	*0.95 ± 0.04*	0.91 ± 0.03	0.92 ± 0.04	0.80 ± 0.07
Wdbc	0.86 ± 0.08	0.85 ± 0.09	0.86 ± 0.08	0.91 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	*0.93 ± 0.04*	0.91 ± 0.05	0.78 ± 0.07
Weathernominal	0.40 ± 0.70	0.40 ± 0.52	0.30 ± 0.67	0.40 ± 0.70	0.50 ± 0.53	0.40 ± 0.52	*0.60 ± 0.52*	*0.60 ± 0.52*	*0.60 ± 0.52*	0.20 ± 0.63
Weathernumeric	0.30 ± 0.67	0.50 ± 0.53	0.50 ± 0.71	*0.70 ± 0.48*	0.50 ± 0.53	0.20 ± 0.79	0.20 ± 0.79	*0.70 ± 0.48*	*0.70 ± 0.48*	0.20 ± 0.63
Website-phishingdata	0.67 ± 0.04	0.71 ± 0.04	*0.84 ± 0.04*	0.79 ± 0.05	0.73 ± 0.02	0.74 ± 0.02	0.78 ± 0.04	0.80 ± 0.04	0.82 ± 0.04	0.66 ± 0.03
Wholesalecustomersdata	0.76 ± 0.05	0.75 ± 0.09	0.78 ± 0.06	0.72 ± 0.11	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	*0.81 ± 0.07*	0.79 ± 0.09	0.78 ± 0.08
Wifi-localization	0.90 ± 0.02	*0.98 ± 0.01*	0.96 ± 0.01	*0.98 ± 0.01*	0.72 ± 0.03	0.74 ± 0.03	0.74 ± 0.03	0.97 ± 0.01	0.97 ± 0.01	0.72 ± 0.06
Wilt	0.00 ± 0.00	0.30 ± 0.08	*0.82 ± 0.05*	0.42 ± 0.07	0.06 ± 0.04	0.06 ± 0.04	0.06 ± 0.04	0.75 ± 0.13	0.79 ± 0.08	0.01 ± 0.03
Wine-quality-red	0.36 ± 0.06	0.31 ± 0.06	0.39 ± 0.05	*0.45 ± 0.04*	0.39 ± 0.04	0.39 ± 0.04	0.40 ± 0.04	0.36 ± 0.06	0.36 ± 0.04	0.24 ± 0.05
Wine-quality-white	0.27 ± 0.04	0.22 ± 0.02	0.38 ± 0.04	*0.49 ± 0.04*	0.30 ± 0.04	0.32 ± 0.04	0.33 ± 0.04	0.28 ± 0.04	0.27 ± 0.03	0.13 ± 0.03
Wine	0.94 ± 0.09	0.95 ± 0.08	0.91 ± 0.08	0.92 ± 0.06	*0.98 ± 0.04*	0.97 ± 0.04	0.97 ± 0.04	0.96 ± 0.06	0.96 ± 0.06	0.65 ± 0.14
Yamilnaduelectricty	0.15 ± 0.01	0.00 ± 0.00	0.09 ± 0.01	*0.17 ± 0.01*	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.01 ± 0.01
Yeast	0.27 ± 0.04	0.45 ± 0.03	0.43 ± 0.06	0.38 ± 0.03	*0.48 ± 0.06*	*0.48 ± 0.06*	*0.48 ± 0.06*	0.47 ± 0.05	0.46 ± 0.04	0.18 ± 0.05
Youtobe-kabita-preprocessing	0.22 ± 0.03	0.14 ± 0.01	0.23 ± 0.02	0.21 ± 0.02	*0.27 ± 0.03*	0.26 ± 0.03	0.25 ± 0.02	0.22 ± 0.01	0.23 ± 0.02	0.21 ± 0.02
Youtobe-nisha-preprocessing	0.27 ± 0.02	0.22 ± 0.02	0.28 ± 0.01	0.25 ± 0.02	*0.31 ± 0.02*	0.29 ± 0.02	0.29 ± 0.03	0.27 ± 0.02	0.26 ± 0.03	0.20 ± 0.02
Z-alizadehsani	0.00 ± 0.00	0.54 ± 0.15	0.47 ± 0.21	0.50 ± 0.13	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	*0.57 ± 0.18*	0.56 ± 0.17	0.30 ± 0.21
Zoo	0.88 ± 0.12	*0.95 ± 0.08*	0.90 ± 0.12	*0.95 ± 0.08*	0.92 ± 0.11	0.92 ± 0.11	0.92 ± 0.11	0.94 ± 0.09	0.94 ± 0.09	0.65 ± 0.14
Average k (rank)	0.4171 (9)	0.5010 (5)	0.5910 (2)	0.5595 (4)	0.4182 (8)	0.4412 (7)	0.4528 (6)	0.5920 (1)	0.5898 (3)	0.4004 (10)
Average k std (rank)	0.0828 (4)	0.1118 (7)	0.1067 (6)	0.1151 (9)	0.0694 (1)	0.0798 (2)	0.0808 (3)	0.1150 (8)	0.1157 (10)	0.1061 (5)

Table 8: The detailed experimental results on root mean squared error and standard deviation.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Abalone	0.41 ± 0.01	0.46 ± 0.02	0.45 ± 0.01	0.53 ± 0.02	0.47 ± 0.02	0.47 ± 0.02	0.47 ± 0.02	0.39 ± 0.01	0.39 ± 0.01	0.51 ± 0.01
Absenteeism-at-work	0.32 ± 0.01	0.33 ± 0.04	0.03 ± 0.05	0.40 ± 0.04	0.39 ± 0.04	0.39 ± 0.04	0.39 ± 0.04	0.14 ± 0.03	0.15 ± 0.04	0.21 ± 0.04
Acute-inflammatioactn	0.25 ± 0.02	0.15 ± 0.03	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.45 ± 0.10
Acute-nephritis	0.22 ± 0.01	0.14 ± 0.13	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.22 ± 0.20
Adult	0.40 ± 0.00	0.40 ± 0.03	0.38 ± 0.03	0.46 ± 0.04	0.41 ± 0.05	0.40 ± 0.05	0.41 ± 0.05	0.38 ± 0.03	0.39 ± 0.03	0.44 ± 0.01
Aggregation	0.14 ± 0.02	0.02 ± 0.01	0.02 ± 0.03	0.01 ± 0.02	0.01 ± 0.02	0.01 ± 0.03	0.01 ± 0.03	0.06 ± 0.02	0.05 ± 0.02	0.34 ± 0.01
Algerianforest	0.33 ± 0.03	0.19 ± 0.14	0.16 ± 0.13	0.33 ± 0.10	0.64 ± 0.01	0.62 ± 0.04	0.62 ± 0.04	0.14 ± 0.10	0.12 ± 0.11	0.09 ± 0.15
Annealing	0.26 ± 0.00	0.31 ± 0.03	0.11 ± 0.04	0.20 ± 0.02	0.21 ± 0.03	0.18 ± 0.03	0.18 ± 0.02	0.17 ± 0.04	0.16 ± 0.04	0.26 ± 0.00
Arrhythmia	0.22 ± 0.00	0.24 ± 0.02	0.22 ± 0.02	0.26 ± 0.01	0.24 ± 0.01	0.23 ± 0.01	0.23 ± 0.01	0.20 ± 0.02	0.23 ± 0.03	0.25 ± 0.01
Au1-1000	0.43 ± 0.00	0.43 ± 0.01	0.43 ± 0.04	0.51 ± 0.04	0.51 ± 0.00	0.51 ± 0.00	0.51 ± 0.01	0.53 ± 0.03	0.51 ± 0.04	0.51 ± 0.00
Au4-2500	0.42 ± 0.00	0.43 ± 0.01	0.41 ± 0.04	0.49 ± 0.03	0.51 ± 0.02	0.51 ± 0.02	0.51 ± 0.02	0.44 ± 0.03	0.44 ± 0.03	0.56 ± 0.03
Au6-1000	0.33 ± 0.00	0.34 ± 0.00	0.43 ± 0.01	0.46 ± 0.01	0.44 ± 0.00	0.44 ± 0.00	0.44 ± 0.00	0.43 ± 0.01	0.43 ± 0.01	0.43 ± 0.01
Au6-250-drift-au6-cd1-500	0.33 ± 0.00	0.34 ± 0.01	0.43 ± 0.01	0.45 ± 0.01	0.44 ± 0.00	0.44 ± 0.00	0.44 ± 0.00	0.42 ± 0.01	0.42 ± 0.01	0.43 ± 0.01
Au6-cd1-400	0.32 ± 0.00	0.34 ± 0.01	0.37 ± 0.02	0.45 ± 0.02	0.42 ± 0.00	0.42 ± 0.00	0.42 ± 0.00	0.40 ± 0.02	0.40 ± 0.01	0.41 ± 0.02
Au7-300-drift-au7-cpd1-800	0.39 ± 0.00	0.39 ± 0.01	0.46 ± 0.02	0.52 ± 0.02	0.54 ± 0.01	0.54 ± 0.01	0.54 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	0.55 ± 0.01
Au7-700	0.36 ± 0.00	0.36 ± 0.01	0.41 ± 0.02	0.51 ± 0.02	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.41 ± 0.01	0.41 ± 0.01	0.47 ± 0.02
Au7-cpd1-500	0.38 ± 0.00	0.38 ± 0.01	0.40 ± 0.02	0.50 ± 0.02	0.50 ± 0.00	0.50 ± 0.01	0.50 ± 0.01	0.45 ± 0.02	0.45 ± 0.03	0.49 ± 0.02
Audiology-std	0.21 ± 0.00	0.16 ± 0.03	0.15 ± 0.03	0.16 ± 0.04	0.20 ± 0.02	0.19 ± 0.03	0.18 ± 0.03	0.12 ± 0.02	0.13 ± 0.02	0.24 ± 0.00
Audit-risk	0.30 ± 0.02	0.22 ± 0.05	0.01 ± 0.04	0.14 ± 0.06	0.10 ± 0.08	0.09 ± 0.08	0.07 ± 0.08	0.15 ± 0.05	0.16 ± 0.05	0.00 ± 0.00
Autism-adolescent-data	0.39 ± 0.03	0.11 ± 0.07	0.00 ± 0.00	0.27 ± 0.15	0.12 ± 0.16	0.15 ± 0.16	0.12 ± 0.16	0.23 ± 0.18	0.23 ± 0.18	0.00 ± 0.00
Autism-adult-data	0.34 ± 0.01	0.12 ± 0.06	0.00 ± 0.00	0.21 ± 0.08	0.05 ± 0.07	0.03 ± 0.06	0.04 ± 0.07	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00
Autism-child-data	0.41 ± 0.01	0.10 ± 0.04	0.00 ± 0.00	0.33 ± 0.06	0.02 ± 0.06	0.00 ± 0.00	0.00 ± 0.00	0.03 ± 0.04	0.04 ± 0.04	0.00 ± 0.00
Autos	0.26 ± 0.01	0.31 ± 0.05	0.20 ± 0.04	0.25 ± 0.06	0.43 ± 0.01	0.43 ± 0.02	0.43 ± 0.02	0.23 ± 0.03	0.23 ± 0.03	0.33 ± 0.04
Avila	0.14 ± 0.00	0.29 ± 0.00	0.09 ± 0.01	0.20 ± 0.01	0.23 ± 0.01	0.22 ± 0.01	0.22 ± 0.01	0.21 ± 0.00	0.21 ± 0.01	0.22 ± 0.01
Balance-scale	0.34 ± 0.01	0.28 ± 0.01	0.37 ± 0.03	0.33 ± 0.04	0.25 ± 0.02	0.22 ± 0.01	0.21 ± 0.02	0.20 ± 0.03	0.21 ± 0.03	0.54 ± 0.03
Balloons	0.42 ± 0.11	0.50 ± 0.03	0.51 ± 0.19	0.37 ± 0.10	0.38 ± 0.41	0.24 ± 0.40	0.24 ± 0.40	0.38 ± 0.33	0.37 ± 0.34	0.65 ± 0.37
Bank	0.31 ± 0.00	0.35 ± 0.01	0.31 ± 0.01	0.37 ± 0.02	0.32 ± 0.01	0.32 ± 0.01	0.32 ± 0.02	0.30 ± 0.01	0.30 ± 0.02	0.34 ± 0.01
Blood	0.42 ± 0.01	0.43 ± 0.02	0.40 ± 0.02	0.53 ± 0.04	0.46 ± 0.02	0.46 ± 0.02	0.46 ± 0.02	0.39 ± 0.01	0.39 ± 0.01	0.49 ± 0.02
Breast-cancer-wisc-diag	0.28 ± 0.02	0.24 ± 0.11	0.24 ± 0.10	0.20 ± 0.05	0.13 ± 0.08	0.12 ± 0.08	0.12 ± 0.07	0.17 ± 0.05	0.17 ± 0.08	0.32 ± 0.06
Breast-cancer-wisc-prog	0.42 ± 0.02	0.52 ± 0.10	0.47 ± 0.08	0.51 ± 0.10	0.45 ± 0.07	0.46 ± 0.10	0.46 ± 0.10	0.46 ± 0.09	0.47 ± 0.10	0.54 ± 0.07
Breast-cancer-wisc	0.20 ± 0.04	0.20 ± 0.05	0.23 ± 0.07	0.19 ± 0.09	0.15 ± 0.08	0.15 ± 0.09	0.17 ± 0.07	0.19 ± 0.10	0.19 ± 0.07	0.26 ± 0.09
Breast-cancer	0.43 ± 0.02	0.45 ± 0.05	0.46 ± 0.03	0.54 ± 0.07	0.51 ± 0.05	0.52 ± 0.06	0.53 ± 0.05	0.48 ± 0.06	0.51 ± 0.04	0.54 ± 0.05
Breast-tissue	0.29 ± 0.03	0.29 ± 0.08	0.30 ± 0.09	0.28 ± 0.10	0.37 ± 0.08	0.32 ± 0.09	0.29 ± 0.13	0.28 ± 0.04	0.29 ± 0.05	0.39 ± 0.04
Bupa	0.47 ± 0.01	0.51 ± 0.04	0.54 ± 0.06	0.62 ± 0.04	0.64 ± 0.02	0.64 ± 0.02	0.64 ± 0.02	0.47 ± 0.04	0.47 ± 0.04	0.67 ± 0.06
Caesarian	0.46 ± 0.05	0.47 ± 0.10	0.49 ± 0.05	0.58 ± 0.12	0.67 ± 0.05	0.61 ± 0.09	0.63 ± 0.12	0.60 ± 0.15	0.60 ± 0.13	0.70 ± 0.15
Car	0.32 ± 0.00	0.23 ± 0.01	0.10 ± 0.03	0.12 ± 0.02	0.11 ± 0.03	0.09 ± 0.03	0.09 ± 0.03	0.15 ± 0.02	0.15 ± 0.02	0.39 ± 0.00
Cardiotocography-10classes	0.24 ± 0.00	0.22 ± 0.01	0.17 ± 0.01	0.21 ± 0.01	0.19 ± 0.01	0.19 ± 0.01	0.18 ± 0.01	0.17 ± 0.01	0.18 ± 0.01	0.32 ± 0.01
Cardiotocography-3classes	0.31 ± 0.00	0.33 ± 0.03	0.21 ± 0.04	0.23 ± 0.03	0.24 ± 0.03	0.23 ± 0.03	0.23 ± 0.03	0.21 ± 0.02	0.20 ± 0.03	0.35 ± 0.03
Cervical-cancer	0.38 ± 0.03	0.23 ± 0.17	0.30 ± 0.22	0.26 ± 0.22	0.54 ± 0.03	0.51 ± 0.07	0.51 ± 0.07	0.16 ± 0.16	0.16 ± 0.16	0.46 ± 0.10
Chemicalcomposionofceramic	0.25 ± 0.02	0.00 ± 0.00	0.04 ± 0.11	0.01 ± 0.00	0.74 ± 0.02	0.74 ± 0.02	0.74 ± 0.02	0.01 ± 0.01	0.01 ± 0.00	0.00 ± 0.00
Chess-krvk	0.22 ± 0.00	0.21 ± 0.00	0.21 ± 0.01	0.23 ± 0.01	0.24 ± 0.01	0.24 ± 0.01	0.24 ± 0.01	0.20 ± 0.00	0.20 ± 0.00	0.29 ± 0.00
Chess-krvkp	0.47 ± 0.00	0.32 ± 0.02	0.07 ± 0.04	0.23 ± 0.02	0.11 ± 0.04	0.08 ± 0.04	0.09 ± 0.03	0.07 ± 0.03	0.07 ± 0.04	0.58 ± 0.01
Congressional-voting	0.48 ± 0.00	0.48 ± 0.04	0.49 ± 0.02	0.50 ± 0.05	0.62 ± 0.03	0.62 ± 0.04	0.63 ± 0.05	0.49 ± 0.04	0.49 ± 0.04	0.61 ± 0.02
Conn-bench-sonar-mines-rocks	0.46 ± 0.01	0.53 ± 0.07	0.52 ± 0.06	0.35 ± 0.10	0.39 ± 0.10	0.33 ± 0.14	0.31 ± 0.14	0.38 ± 0.14	0.36 ± 0.14	0.60 ± 0.09

Table 8: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Conn-bench-vowel-deterding	0.22 ± 0.00	0.20 ± 0.01	0.18 ± 0.03	0.02 ± 0.03	0.10 ± 0.02	0.06 ± 0.03	0.04 ± 0.04	0.15 ± 0.02	0.16 ± 0.02	0.34 ± 0.01
Connect-4	0.43 ± 0.00	0.48 ± 0.01	0.41 ± 0.02	0.53 ± 0.03	0.51 ± 0.00	0.49 ± 0.01	0.48 ± 0.02	0.47 ± 0.02	0.47 ± 0.02	0.51 ± 0.00
Connectionist	0.46 ± 0.01	0.53 ± 0.10	0.49 ± 0.07	0.35 ± 0.08	0.62 ± 0.07	0.54 ± 0.13	0.49 ± 0.19	0.36 ± 0.11	0.36 ± 0.11	0.61 ± 0.06
Contrac	0.45 ± 0.00	0.46 ± 0.02	0.47 ± 0.02	0.61 ± 0.02	0.55 ± 0.01	0.55 ± 0.02	0.55 ± 0.02	0.43 ± 0.01	0.43 ± 0.01	0.59 ± 0.01
Covid-19	0.38 ± 0.09	0.35 ± 0.19	0.33 ± 0.31	0.40 ± 0.27	0.39 ± 0.35	0.42 ± 0.37	0.34 ± 0.37	0.31 ± 0.29	0.31 ± 0.30	0.20 ± 0.32
Credit-approval	0.42 ± 0.01	0.43 ± 0.04	0.36 ± 0.04	0.42 ± 0.07	0.38 ± 0.07	0.38 ± 0.06	0.39 ± 0.06	0.35 ± 0.04	0.37 ± 0.04	0.38 ± 0.06
Crowdsource	0.22 ± 0.00	0.22 ± 0.02	0.19 ± 0.01	0.12 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.15 ± 0.02	0.15 ± 0.02	0.27 ± 0.01
Crx	0.42 ± 0.01	0.43 ± 0.03	0.33 ± 0.04	0.43 ± 0.06	0.66 ± 0.03	0.66 ± 0.03	0.65 ± 0.04	0.38 ± 0.05	0.37 ± 0.03	0.38 ± 0.06
Cryother	0.34 ± 0.05	0.31 ± 0.16	0.20 ± 0.15	0.26 ± 0.18	0.39 ± 0.18	0.37 ± 0.18	0.37 ± 0.18	0.28 ± 0.13	0.32 ± 0.12	0.42 ± 0.12
Cylinder-bands	0.46 ± 0.01	0.54 ± 0.06	0.50 ± 0.06	0.55 ± 0.06	0.49 ± 0.04	0.46 ± 0.05	0.43 ± 0.04	0.48 ± 0.05	0.48 ± 0.04	0.58 ± 0.04
Dbworld-bodies	0.49 ± 0.01	0.46 ± 0.20	0.36 ± 0.25	0.62 ± 0.08	0.67 ± 0.04	0.67 ± 0.04	0.59 ± 0.11	/	/	0.28 ± 0.25
Dbworld-bodies-stemmed	0.49 ± 0.01	0.45 ± 0.18	0.28 ± 0.22	0.58 ± 0.12	0.67 ± 0.04	0.64 ± 0.07	0.50 ± 0.12	/	/	0.44 ± 0.27
Dbworld-subjects	0.49 ± 0.01	0.28 ± 0.15	0.44 ± 0.10	0.42 ± 0.15	0.67 ± 0.04	0.67 ± 0.04	0.67 ± 0.04	0.25 ± 0.20	0.25 ± 0.23	0.60 ± 0.10
Dbworld-subjects-stemmed	0.49 ± 0.01	0.27 ± 0.16	0.39 ± 0.17	0.37 ± 0.14	0.67 ± 0.04	0.67 ± 0.04	0.67 ± 0.04	0.26 ± 0.20	0.27 ± 0.20	0.62 ± 0.08
Dermatology	0.27 ± 0.00	0.07 ± 0.05	0.09 ± 0.06	0.10 ± 0.07	0.07 ± 0.06	0.06 ± 0.06	0.06 ± 0.07	0.07 ± 0.03	0.07 ± 0.03	0.41 ± 0.01
Diabetes	0.43 ± 0.02	0.41 ± 0.04	0.44 ± 0.05	0.54 ± 0.04	0.59 ± 0.00	0.59 ± 0.00	0.59 ± 0.00	0.42 ± 0.03	0.42 ± 0.04	0.53 ± 0.05
Diabetic	0.47 ± 0.01	0.65 ± 0.02	0.51 ± 0.03	0.62 ± 0.04	0.65 ± 0.04	0.64 ± 0.05	0.63 ± 0.05	0.43 ± 0.04	0.43 ± 0.04	0.68 ± 0.04
Divorce	0.15 ± 0.06	0.10 ± 0.13	0.17 ± 0.14	0.09 ± 0.11	0.10 ± 0.13	0.10 ± 0.13	0.10 ± 0.13	0.08 ± 0.10	0.08 ± 0.11	0.16 ± 0.15
Dota2train	0.50 ± 0.00	0.52 ± 0.02	0.60 ± 0.04	0.62 ± 0.03	0.69 ± 0.00	0.68 ± 0.03	0.69 ± 0.02	0.65 ± 0.07	0.64 ± 0.08	0.67 ± 0.03
Dow-jones-index	0.47 ± 0.01	0.55 ± 0.03	0.46 ± 0.03	0.67 ± 0.05	0.69 ± 0.00	0.69 ± 0.00	0.69 ± 0.00	0.53 ± 0.03	0.54 ± 0.03	0.66 ± 0.02
Dry-bean-dataset	0.14 ± 0.00	0.16 ± 0.01	0.13 ± 0.01	0.12 ± 0.01	0.33 ± 0.01	0.33 ± 0.01	0.33 ± 0.01	0.13 ± 0.01	0.13 ± 0.01	0.30 ± 0.01
Early-stage-diabetes-data-upload	0.39 ± 0.01	0.32 ± 0.05	0.19 ± 0.06	0.11 ± 0.09	0.22 ± 0.11	0.20 ± 0.11	0.18 ± 0.12	0.15 ± 0.07	0.16 ± 0.08	0.42 ± 0.05
Echocardiogram	0.43 ± 0.03	0.40 ± 0.15	0.42 ± 0.11	0.49 ± 0.11	0.40 ± 0.15	0.41 ± 0.17	0.41 ± 0.17	0.42 ± 0.11	0.42 ± 0.12	0.37 ± 0.09
Ecoli	0.24 ± 0.01	0.16 ± 0.03	0.18 ± 0.05	0.22 ± 0.04	0.18 ± 0.03	0.18 ± 0.03	0.18 ± 0.03	0.17 ± 0.02	0.18 ± 0.02	0.29 ± 0.03
Eegeyesate	0.46 ± 0.00	0.70 ± 0.06	0.38 ± 0.01	0.40 ± 0.03	0.67 ± 0.00	0.67 ± 0.00	0.67 ± 0.00	0.50 ± 0.01	0.50 ± 0.02	0.61 ± 0.01
Electrical	0.35 ± 0.04	0.14 ± 0.07	0.00 ± 0.00	0.27 ± 0.12	0.41 ± 0.09	0.41 ± 0.08	0.40 ± 0.08	0.08 ± 0.07	0.08 ± 0.07	0.02 ± 0.06
Energy-y1	0.31 ± 0.00	0.30 ± 0.02	0.12 ± 0.03	0.40 ± 0.03	0.27 ± 0.02	0.26 ± 0.02	0.26 ± 0.04	0.24 ± 0.01	0.25 ± 0.02	0.32 ± 0.02
Energy-y2	0.29 ± 0.01	0.27 ± 0.03	0.22 ± 0.03	0.40 ± 0.03	0.26 ± 0.03	0.25 ± 0.02	0.24 ± 0.02	0.21 ± 0.02	0.21 ± 0.03	0.28 ± 0.04
Extentionofz-alizadehsani	0.42 ± 0.01	0.23 ± 0.07	0.02 ± 0.06	0.30 ± 0.08	0.54 ± 0.01	0.54 ± 0.01	0.54 ± 0.01	0.09 ± 0.09	0.09 ± 0.09	0.36 ± 0.07
Fertility	0.32 ± 0.05	0.33 ± 0.05	0.36 ± 0.07	0.38 ± 0.17	0.34 ± 0.06	0.31 ± 0.12	0.29 ± 0.17	0.29 ± 0.13	0.30 ± 0.15	0.34 ± 0.06
First-order	0.29 ± 0.00	0.82 ± 0.02	0.00 ± 0.00	0.06 ± 0.05	0.31 ± 0.01	0.21 ± 0.04	0.13 ± 0.06	0.02 ± 0.03	0.02 ± 0.03	0.30 ± 0.03
Flags	0.29 ± 0.01	0.33 ± 0.02	0.30 ± 0.04	0.36 ± 0.03	0.34 ± 0.03	0.34 ± 0.04	0.34 ± 0.05	0.33 ± 0.04	0.32 ± 0.03	0.33 ± 0.03
Foresttypes	0.28 ± 0.01	0.13 ± 0.09	0.12 ± 0.10	0.13 ± 0.08	0.59 ± 0.01	0.58 ± 0.02	0.58 ± 0.02	0.10 ± 0.07	0.10 ± 0.07	0.29 ± 0.07
Garments-worker-productivity	0.37 ± 0.00	0.41 ± 0.01	0.41 ± 0.02	0.54 ± 0.01	0.51 ± 0.01	0.51 ± 0.01	0.50 ± 0.01	0.39 ± 0.01	0.39 ± 0.01	0.52 ± 0.01
Gender-name-dataset	0.49 ± 0.00	0.61 ± 0.01	0.49 ± 0.00	0.48 ± 0.01	0.60 ± 0.01	0.60 ± 0.02	0.60 ± 0.02	0.49 ± 0.00	0.49 ± 0.00	0.63 ± 0.01
Gesture-a1-raw	0.20 ± 0.01	0.29 ± 0.02	0.18 ± 0.02	0.12 ± 0.02	0.49 ± 0.00	0.49 ± 0.00	0.49 ± 0.00	0.17 ± 0.02	0.17 ± 0.01	0.11 ± 0.03
Gesture-a1-va3	0.31 ± 0.01	0.38 ± 0.02	0.35 ± 0.03	0.32 ± 0.02	0.49 ± 0.00	0.49 ± 0.00	0.49 ± 0.00	0.30 ± 0.01	0.31 ± 0.02	0.39 ± 0.01
Gesture-a2-raw	0.21 ± 0.00	0.33 ± 0.02	0.20 ± 0.03	0.14 ± 0.05	0.49 ± 0.00	0.49 ± 0.00	0.49 ± 0.00	0.18 ± 0.02	0.18 ± 0.02	0.14 ± 0.03
Gesture-a2-va3	0.33 ± 0.01	0.44 ± 0.04	0.40 ± 0.01	0.36 ± 0.02	0.49 ± 0.00	0.49 ± 0.00	0.49 ± 0.00	0.35 ± 0.02	0.35 ± 0.02	0.43 ± 0.02
Gesture-a3-raw	0.18 ± 0.01	0.35 ± 0.01	0.16 ± 0.01	0.12 ± 0.02	0.51 ± 0.00	0.51 ± 0.00	0.51 ± 0.00	0.19 ± 0.01	0.19 ± 0.02	0.14 ± 0.02
Gesture-a3-va3	0.33 ± 0.00	0.43 ± 0.01	0.30 ± 0.02	0.20 ± 0.02	0.51 ± 0.00	0.51 ± 0.00	0.51 ± 0.00	0.32 ± 0.01	0.32 ± 0.01	0.44 ± 0.01
Gesture-b1-raw	0.18 ± 0.01	0.38 ± 0.02	0.15 ± 0.03	0.13 ± 0.04	0.45 ± 0.00	0.45 ± 0.00	0.45 ± 0.00	0.18 ± 0.02	0.17 ± 0.01	0.17 ± 0.02
Gesture-b1-va3	0.33 ± 0.01	0.49 ± 0.03	0.34 ± 0.02	0.19 ± 0.03	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.33 ± 0.01	0.33 ± 0.02	0.48 ± 0.02
Gesture-b3-raw	0.23 ± 0.01	0.36 ± 0.02	0.18 ± 0.03	0.13 ± 0.01	0.52 ± 0.00	0.52 ± 0.00	0.52 ± 0.00	0.20 ± 0.04	0.20 ± 0.03	0.13 ± 0.03

Table 8: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Gesture-b3-va3	0.37 ± 0.01	0.45 ± 0.02	0.40 ± 0.02	0.30 ± 0.02	0.52 ± 0.00	0.52 ± 0.00	0.52 ± 0.00	0.35 ± 0.02	0.37 ± 0.02	0.49 ± 0.02
Gesture-c1-raw	0.23 ± 0.01	0.34 ± 0.01	0.20 ± 0.02	0.15 ± 0.02	0.55 ± 0.00	0.55 ± 0.00	0.55 ± 0.00	0.20 ± 0.03	0.20 ± 0.03	0.15 ± 0.03
Gesture-c1-va3	0.35 ± 0.01	0.41 ± 0.02	0.39 ± 0.02	0.34 ± 0.02	0.55 ± 0.00	0.55 ± 0.00	0.55 ± 0.00	0.34 ± 0.02	0.35 ± 0.01	0.48 ± 0.02
Gesture-c3-raw	0.22 ± 0.01	0.36 ± 0.02	0.21 ± 0.03	0.16 ± 0.03	0.54 ± 0.00	0.54 ± 0.00	0.54 ± 0.00	0.21 ± 0.02	0.23 ± 0.01	0.15 ± 0.02
Gesture-c3-va3	0.36 ± 0.01	0.44 ± 0.01	0.40 ± 0.02	0.39 ± 0.02	0.54 ± 0.00	0.54 ± 0.00	0.54 ± 0.00	0.36 ± 0.02	0.37 ± 0.02	0.48 ± 0.01
Glass	0.29 ± 0.01	0.36 ± 0.03	0.31 ± 0.04	0.31 ± 0.05	0.30 ± 0.04	0.30 ± 0.04	0.31 ± 0.04	0.28 ± 0.03	0.28 ± 0.02	0.38 ± 0.04
Go-track-tracks	0.37 ± 0.04	0.38 ± 0.06	0.30 ± 0.14	0.24 ± 0.18	0.52 ± 0.11	0.52 ± 0.11	0.52 ± 0.11	0.32 ± 0.09	0.32 ± 0.09	0.48 ± 0.14
Haberman-survival	0.43 ± 0.01	0.43 ± 0.04	0.43 ± 0.02	0.57 ± 0.06	0.52 ± 0.05	0.54 ± 0.03	0.54 ± 0.03	0.43 ± 0.03	0.43 ± 0.03	0.52 ± 0.03
Hayes-roth	0.34 ± 0.02	0.36 ± 0.02	0.28 ± 0.05	0.21 ± 0.05	0.28 ± 0.11	0.35 ± 0.06	0.36 ± 0.06	0.34 ± 0.10	0.34 ± 0.10	0.61 ± 0.03
Hcc-data	0.46 ± 0.01	0.53 ± 0.07	0.54 ± 0.06	0.60 ± 0.07	0.62 ± 0.02	0.62 ± 0.02	0.62 ± 0.02	0.54 ± 0.11	0.56 ± 0.09	0.54 ± 0.12
Hcvdat	0.19 ± 0.01	0.17 ± 0.04	0.17 ± 0.03	0.19 ± 0.02	0.23 ± 0.01	0.23 ± 0.01	0.23 ± 0.01	0.14 ± 0.03	0.16 ± 0.03	0.21 ± 0.02
Heart-cleveland	0.33 ± 0.01	0.34 ± 0.04	0.40 ± 0.04	0.42 ± 0.03	0.41 ± 0.04	0.40 ± 0.05	0.41 ± 0.05	0.38 ± 0.03	0.38 ± 0.04	0.43 ± 0.03
Heart-hungarian	0.39 ± 0.03	0.39 ± 0.08	0.41 ± 0.07	0.46 ± 0.09	0.42 ± 0.09	0.41 ± 0.08	0.40 ± 0.09	0.42 ± 0.08	0.42 ± 0.06	0.45 ± 0.11
Heart-switzerland	0.37 ± 0.01	0.40 ± 0.03	0.45 ± 0.05	0.51 ± 0.05	0.49 ± 0.03	0.49 ± 0.02	0.48 ± 0.03	0.42 ± 0.05	0.42 ± 0.04	0.52 ± 0.05
Heart-va	0.39 ± 0.01	0.42 ± 0.03	0.48 ± 0.02	0.50 ± 0.03	0.51 ± 0.04	0.52 ± 0.03	0.53 ± 0.04	0.47 ± 0.03	0.48 ± 0.03	0.54 ± 0.04
Heart-failure-clinical-records-dataset	0.44 ± 0.01	0.41 ± 0.04	0.41 ± 0.06	0.58 ± 0.06	0.57 ± 0.01	0.57 ± 0.01	0.57 ± 0.01	0.48 ± 0.05	0.45 ± 0.09	0.37 ± 0.08
Hepatitis	0.38 ± 0.02	0.37 ± 0.10	0.41 ± 0.12	0.41 ± 0.07	0.40 ± 0.08	0.35 ± 0.19	0.35 ± 0.19	0.39 ± 0.11	0.42 ± 0.12	0.50 ± 0.07
Hill-valley	0.51 ± 0.01	0.70 ± 0.05	0.50 ± 0.00	0.69 ± 0.03	0.69 ± 0.05	0.69 ± 0.05	0.68 ± 0.05	0.48 ± 0.02	0.49 ± 0.02	0.72 ± 0.03
Hiv1625data	0.25 ± 0.02	0.22 ± 0.04	0.27 ± 0.03	0.27 ± 0.02	0.28 ± 0.04	0.25 ± 0.04	0.25 ± 0.05	0.20 ± 0.05	0.24 ± 0.05	0.45 ± 0.02
Hiv746data	0.30 ± 0.03	0.26 ± 0.05	0.37 ± 0.04	0.34 ± 0.04	0.32 ± 0.06	0.30 ± 0.05	0.29 ± 0.06	0.25 ± 0.05	0.26 ± 0.06	0.43 ± 0.06
Horse-colic	0.43 ± 0.01	0.47 ± 0.06	0.34 ± 0.08	0.48 ± 0.08	0.40 ± 0.07	0.39 ± 0.07	0.38 ± 0.08	0.44 ± 0.05	0.42 ± 0.10	0.42 ± 0.06
Htru	0.16 ± 0.01	0.23 ± 0.01	0.14 ± 0.02	0.17 ± 0.01	0.30 ± 0.00	0.29 ± 0.00	0.29 ± 0.00	0.13 ± 0.01	0.13 ± 0.01	0.15 ± 0.01
Hypothyroid	0.18 ± 0.00	0.14 ± 0.01	0.04 ± 0.02	0.21 ± 0.02	0.19 ± 0.00	0.19 ± 0.01	0.19 ± 0.01	0.15 ± 0.04	0.17 ± 0.04	0.14 ± 0.02
Ibeacon-rssi-labeled	0.09 ± 0.00	0.10 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.13 ± 0.00
Ilpd-indian-liver	0.43 ± 0.01	0.65 ± 0.04	0.47 ± 0.03	0.59 ± 0.05	0.54 ± 0.01	0.54 ± 0.02	0.54 ± 0.02	0.42 ± 0.01	0.42 ± 0.01	0.58 ± 0.03
Image-segmentation	0.23 ± 0.01	0.24 ± 0.03	0.16 ± 0.08	0.19 ± 0.04	0.19 ± 0.02	0.19 ± 0.04	0.19 ± 0.04	0.14 ± 0.03	0.15 ± 0.03	0.34 ± 0.04
Immunotherapy	0.38 ± 0.02	0.40 ± 0.08	0.37 ± 0.13	0.50 ± 0.21	0.46 ± 0.04	0.46 ± 0.04	0.46 ± 0.04	0.37 ± 0.10	0.39 ± 0.11	0.37 ± 0.16
Impensdata	0.30 ± 0.01	0.26 ± 0.03	0.36 ± 0.00	0.33 ± 0.03	0.40 ± 0.00	0.40 ± 0.00	0.40 ± 0.00	0.29 ± 0.03	0.31 ± 0.04	0.39 ± 0.03
In-vehicle-coupon-recommendation	0.48 ± 0.00	0.47 ± 0.01	0.46 ± 0.01	0.56 ± 0.01	0.56 ± 0.01	0.55 ± 0.01	0.55 ± 0.01	0.52 ± 0.01	0.51 ± 0.01	0.63 ± 0.01
Indian	0.43 ± 0.01	0.65 ± 0.04	0.48 ± 0.04	0.60 ± 0.06	0.53 ± 0.02	0.53 ± 0.01	0.53 ± 0.01	0.43 ± 0.02	0.43 ± 0.02	0.58 ± 0.03
Ionosphere	0.30 ± 0.02	0.39 ± 0.06	0.28 ± 0.06	0.36 ± 0.07	0.23 ± 0.05	0.21 ± 0.09	0.19 ± 0.09	0.29 ± 0.03	0.28 ± 0.05	0.42 ± 0.11
Iris	0.28 ± 0.04	0.14 ± 0.08	0.11 ± 0.12	0.13 ± 0.13	0.11 ± 0.11	0.11 ± 0.12	0.11 ± 0.11	0.10 ± 0.09	0.10 ± 0.09	0.19 ± 0.14
Jain	0.20 ± 0.06	0.18 ± 0.06	0.02 ± 0.07	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.18 ± 0.05	0.18 ± 0.05	0.19 ± 0.11
Jsbach-chorals-harmony	0.26 ± 0.00	0.16 ± 0.01	0.17 ± 0.01	0.14 ± 0.01	0.19 ± 0.02	0.18 ± 0.01	0.17 ± 0.02	0.15 ± 0.02	0.16 ± 0.03	0.20 ± 0.02
Knowledge	0.36 ± 0.02	0.25 ± 0.07	0.15 ± 0.10	0.34 ± 0.06	0.27 ± 0.09	0.20 ± 0.13	0.16 ± 0.13	0.17 ± 0.08	0.17 ± 0.07	0.29 ± 0.08
Lasvegastripadvisorreviews	0.35 ± 0.00	0.39 ± 0.01	0.42 ± 0.01	0.49 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.45 ± 0.01	0.45 ± 0.02	0.49 ± 0.01
Leaf	0.15 ± 0.00	0.12 ± 0.02	0.15 ± 0.02	0.24 ± 0.00	0.24 ± 0.01	0.22 ± 0.01	0.22 ± 0.01	0.15 ± 0.01	0.16 ± 0.01	0.23 ± 0.01
Led-display	0.26 ± 0.00	0.20 ± 0.01	0.21 ± 0.01	0.21 ± 0.01	0.24 ± 0.02	0.24 ± 0.02	0.24 ± 0.02	0.20 ± 0.01	0.21 ± 0.01	0.40 ± 0.00
Lenses	0.33 ± 0.10	0.42 ± 0.12	0.23 ± 0.26	0.19 ± 0.18	0.24 ± 0.31	0.21 ± 0.27	0.21 ± 0.27	0.29 ± 0.24	0.31 ± 0.25	0.39 ± 0.29
Letter	0.17 ± 0.00	0.14 ± 0.00	0.09 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.05 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.25 ± 0.00
Libras	0.21 ± 0.01	0.21 ± 0.03	0.19 ± 0.03	0.13 ± 0.03	0.16 ± 0.02	0.14 ± 0.02	0.13 ± 0.02	0.15 ± 0.02	0.14 ± 0.02	0.32 ± 0.01
Low-res-spect	0.21 ± 0.00	0.21 ± 0.02	0.19 ± 0.02	0.19 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.14 ± 0.02	0.12 ± 0.02	0.13 ± 0.02	0.24 ± 0.02
Lung-cancer	0.44 ± 0.02	0.45 ± 0.14	0.55 ± 0.14	0.48 ± 0.27	0.52 ± 0.20	0.49 ± 0.20	0.53 ± 0.21	0.54 ± 0.11	0.54 ± 0.12	0.58 ± 0.10
Lymphography	0.32 ± 0.01	0.26 ± 0.06	0.33 ± 0.07	0.32 ± 0.13	0.24 ± 0.11	0.26 ± 0.08	0.28 ± 0.07	0.27 ± 0.06	0.27 ± 0.07	0.34 ± 0.06

Table 8: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Magic	0.40 ± 0.01	0.49 ± 0.02	0.38 ± 0.02	0.45 ± 0.03	0.39 ± 0.03	0.39 ± 0.02	0.39 ± 0.02	0.34 ± 0.03	0.36 ± 0.03	0.53 ± 0.04
Mammographic	0.40 ± 0.01	0.41 ± 0.04	0.36 ± 0.02	0.47 ± 0.05	0.41 ± 0.03	0.42 ± 0.03	0.42 ± 0.03	0.37 ± 0.02	0.38 ± 0.03	0.43 ± 0.01
Miniboone	0.36 ± 0.01	0.84 ± 0.01	0.36 ± 0.03	0.40 ± 0.02	0.39 ± 0.03	0.38 ± 0.03	0.37 ± 0.03	0.34 ± 0.02	0.34 ± 0.02	0.43 ± 0.02
Molec-biol-promoter	0.45 ± 0.02	0.27 ± 0.14	0.48 ± 0.10	0.51 ± 0.12	0.40 ± 0.14	0.36 ± 0.17	0.36 ± 0.17	0.42 ± 0.10	0.43 ± 0.10	0.54 ± 0.10
Molec-biol-splice	0.44 ± 0.00	0.20 ± 0.02	0.21 ± 0.03	0.50 ± 0.01	0.29 ± 0.02	0.29 ± 0.02	0.30 ± 0.01	0.29 ± 0.02	0.30 ± 0.02	0.49 ± 0.02
Monks-1	0.40 ± 0.03	0.43 ± 0.08	0.09 ± 0.13	0.54 ± 0.10	0.39 ± 0.22	0.36 ± 0.21	0.38 ± 0.16	0.22 ± 0.22	0.18 ± 0.24	0.51 ± 0.10
Monks-2	0.49 ± 0.02	0.50 ± 0.03	0.44 ± 0.10	0.61 ± 0.09	0.65 ± 0.06	0.62 ± 0.11	0.59 ± 0.10	0.45 ± 0.10	0.44 ± 0.07	0.64 ± 0.04
Monks-3	0.39 ± 0.04	0.32 ± 0.06	0.20 ± 0.16	0.47 ± 0.15	0.26 ± 0.19	0.29 ± 0.17	0.29 ± 0.17	0.32 ± 0.16	0.28 ± 0.12	0.47 ± 0.04
Mushroom	0.34 ± 0.01	0.32 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.12 ± 0.02
Musk-1	0.41 ± 0.01	0.49 ± 0.07	0.37 ± 0.09	0.38 ± 0.06	0.30 ± 0.08	0.27 ± 0.06	0.24 ± 0.06	0.21 ± 0.05	0.21 ± 0.05	0.61 ± 0.07
Musk-2	0.29 ± 0.01	0.37 ± 0.02	0.20 ± 0.03	0.19 ± 0.02	0.20 ± 0.03	0.16 ± 0.03	0.14 ± 0.03	0.10 ± 0.04	0.10 ± 0.04	0.32 ± 0.02
Newdiagnosis	0.25 ± 0.02	0.15 ± 0.04	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.47 ± 0.08
Nursery	0.33 ± 0.00	0.19 ± 0.01	0.04 ± 0.01	0.18 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.10 ± 0.02	0.12 ± 0.02	0.34 ± 0.01
Obesitydataset-raw-and-data-sinthetic	0.28 ± 0.00	0.25 ± 0.01	0.13 ± 0.02	0.23 ± 0.02	0.17 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.12 ± 0.03	0.12 ± 0.01	0.31 ± 0.02
Obs-network-dataset-2-aug27	0.21 ± 0.01	0.36 ± 0.03	0.01 ± 0.03	0.06 ± 0.03	0.11 ± 0.02	0.11 ± 0.02	0.11 ± 0.02	0.11 ± 0.03	0.09 ± 0.03	0.21 ± 0.03
Occupancy-data	0.17 ± 0.02	0.20 ± 0.02	0.11 ± 0.03	0.09 ± 0.03	0.44 ± 0.02	0.42 ± 0.03	0.42 ± 0.03	0.13 ± 0.03	0.12 ± 0.03	0.11 ± 0.03
Occupancy-data2	0.14 ± 0.01	0.19 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.41 ± 0.01	0.38 ± 0.01	0.38 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.08 ± 0.01
Occupancy-data3	0.12 ± 0.01	0.14 ± 0.02	0.07 ± 0.01	0.07 ± 0.02	0.41 ± 0.01	0.39 ± 0.01	0.39 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.08 ± 0.01
Old	0.42 ± 0.00	0.21 ± 0.02	0.20 ± 0.03	0.13 ± 0.03	0.22 ± 0.02	0.22 ± 0.03	0.21 ± 0.03	0.15 ± 0.03	0.15 ± 0.03	0.34 ± 0.02
Online-shoppers-intention	0.34 ± 0.00	0.39 ± 0.01	0.30 ± 0.01	0.43 ± 0.01	0.39 ± 0.00	0.39 ± 0.00	0.40 ± 0.00	0.28 ± 0.01	0.29 ± 0.02	0.34 ± 0.01
Oocytes-merluccius-nucleus-4d	0.43 ± 0.01	0.59 ± 0.04	0.44 ± 0.02	0.53 ± 0.04	0.48 ± 0.03	0.44 ± 0.03	0.43 ± 0.04	0.38 ± 0.03	0.39 ± 0.04	0.57 ± 0.04
Oocytes-merluccius-states-2f	0.25 ± 0.02	0.31 ± 0.04	0.24 ± 0.04	0.24 ± 0.04	0.23 ± 0.04	0.23 ± 0.03	0.22 ± 0.03	0.21 ± 0.02	0.22 ± 0.03	0.34 ± 0.04
Oocytes-trisopterus-nucleus-2f	0.46 ± 0.01	0.64 ± 0.04	0.49 ± 0.05	0.50 ± 0.06	0.43 ± 0.04	0.40 ± 0.04	0.41 ± 0.03	0.36 ± 0.04	0.39 ± 0.04	0.64 ± 0.05
Oocytes-trisopterus-states-5b	0.29 ± 0.03	0.39 ± 0.03	0.26 ± 0.04	0.24 ± 0.05	0.23 ± 0.05	0.22 ± 0.04	0.22 ± 0.04	0.19 ± 0.04	0.19 ± 0.04	0.36 ± 0.03
Optdigits	0.27 ± 0.00	0.13 ± 0.01	0.13 ± 0.01	0.05 ± 0.01	0.23 ± 0.01	0.22 ± 0.01	0.22 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.38 ± 0.00
Optical	0.27 ± 0.00	0.12 ± 0.01	0.14 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.06 ± 0.01	0.38 ± 0.00
Ozone	0.17 ± 0.01	0.53 ± 0.03	0.20 ± 0.01	0.22 ± 0.02	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01	0.18 ± 0.02	0.17 ± 0.02	0.18 ± 0.01
Page-blocks	0.14 ± 0.01	0.19 ± 0.01	0.11 ± 0.01	0.13 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.16 ± 0.01
Parkingbirmingham	0.09 ± 0.01	0.10 ± 0.00	0.00 ± 0.00	0.04 ± 0.01	0.17 ± 0.01	0.16 ± 0.01	0.16 ± 0.01	0.08 ± 0.00	0.08 ± 0.00	0.00 ± 0.00
Parkinsons	0.34 ± 0.03	0.54 ± 0.11	0.42 ± 0.05	0.16 ± 0.11	0.34 ± 0.09	0.29 ± 0.13	0.27 ± 0.12	0.28 ± 0.11	0.26 ± 0.14	0.36 ± 0.11
Pasture	0.41 ± 0.04	0.34 ± 0.24	0.30 ± 0.23	0.33 ± 0.26	0.69 ± 0.02	0.69 ± 0.02	0.69 ± 0.02	0.30 ± 0.24	0.31 ± 0.25	0.43 ± 0.25
Pbc	0.43 ± 0.02	0.43 ± 0.07	0.41 ± 0.07	0.63 ± 0.07	0.62 ± 0.01	0.62 ± 0.01	0.62 ± 0.01	0.50 ± 0.05	0.48 ± 0.03	0.52 ± 0.09
Pen	0.22 ± 0.00	0.16 ± 0.01	0.08 ± 0.01	0.04 ± 0.00	0.42 ± 0.00	0.41 ± 0.00	0.41 ± 0.00	0.10 ± 0.01	0.10 ± 0.00	0.35 ± 0.00
Pendigits	0.22 ± 0.00	0.15 ± 0.01	0.09 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.35 ± 0.01
Pharynx	0.43 ± 0.02	0.46 ± 0.03	0.42 ± 0.01	0.59 ± 0.09	0.49 ± 0.08	0.51 ± 0.10	0.54 ± 0.09	0.49 ± 0.04	0.51 ± 0.13	0.85 ± 0.03
Phishingwebsites	0.44 ± 0.00	0.23 ± 0.01	0.19 ± 0.01	0.14 ± 0.01	0.23 ± 0.01	0.22 ± 0.01	0.22 ± 0.01	0.16 ± 0.01	0.16 ± 0.01	0.33 ± 0.01
Pima	0.43 ± 0.02	0.42 ± 0.04	0.44 ± 0.05	0.54 ± 0.04	0.49 ± 0.04	0.50 ± 0.04	0.50 ± 0.03	0.42 ± 0.03	0.42 ± 0.04	0.53 ± 0.05
Pittsburg-bridges-rel-l	0.40 ± 0.03	0.40 ± 0.08	0.44 ± 0.08	0.40 ± 0.17	0.46 ± 0.10	0.45 ± 0.10	0.42 ± 0.10	0.45 ± 0.09	0.43 ± 0.09	0.43 ± 0.09
Pittsburg-bridges-span	0.41 ± 0.02	0.38 ± 0.07	0.43 ± 0.07	0.51 ± 0.11	0.47 ± 0.10	0.48 ± 0.12	0.48 ± 0.12	0.41 ± 0.05	0.41 ± 0.07	0.54 ± 0.08
Pittsburg-bridges-t-or-d	0.33 ± 0.06	0.30 ± 0.10	0.37 ± 0.13	0.37 ± 0.17	0.34 ± 0.14	0.30 ± 0.17	0.28 ± 0.16	0.29 ± 0.21	0.31 ± 0.22	0.34 ± 0.19
Pittsburg-bridges-type	0.33 ± 0.02	0.32 ± 0.04	0.33 ± 0.04	0.36 ± 0.06	0.39 ± 0.05	0.35 ± 0.08	0.35 ± 0.07	0.32 ± 0.07	0.34 ± 0.07	0.38 ± 0.02
Pittsburg-bridgesmaterial	0.30 ± 0.03	0.30 ± 0.07	0.25 ± 0.08	0.30 ± 0.13	0.30 ± 0.07	0.31 ± 0.06	0.30 ± 0.13	0.30 ± 0.08	0.32 ± 0.08	0.29 ± 0.05
Planning	0.45 ± 0.01	0.49 ± 0.04	0.45 ± 0.01	0.58 ± 0.08	0.53 ± 0.01	0.54 ± 0.05	0.58 ± 0.06	0.56 ± 0.06	0.60 ± 0.08	0.61 ± 0.07
Plant-margin	0.09 ± 0.00	0.05 ± 0.00	0.10 ± 0.00	0.07 ± 0.00	0.06 ± 0.00	0.06 ± 0.01	0.05 ± 0.01	0.05 ± 0.00	0.05 ± 0.00	0.14 ± 0.00

Table 8: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Plant-shape	0.09 ± 0.00	0.09 ± 0.00	0.10 ± 0.00	0.08 ± 0.00	0.10 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.07 ± 0.00	0.07 ± 0.00	0.14 ± 0.00
Plant-texture	0.10 ± 0.00	0.07 ± 0.00	0.09 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.14 ± 0.00
Poker-hand-training-true	0.24 ± 0.00	0.24 ± 0.00	0.27 ± 0.01	0.32 ± 0.00	0.29 ± 0.00	0.29 ± 0.00	0.30 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.32 ± 0.00
Post-operative	0.38 ± 0.03	0.41 ± 0.04	0.38 ± 0.03	0.51 ± 0.06	0.44 ± 0.04	0.45 ± 0.05	0.47 ± 0.07	0.45 ± 0.08	0.48 ± 0.08	0.45 ± 0.05
Primary-tumor	0.23 ± 0.00	0.22 ± 0.02	0.24 ± 0.01	0.28 ± 0.02	0.27 ± 0.03	0.27 ± 0.03	0.27 ± 0.03	0.25 ± 0.02	0.25 ± 0.02	0.31 ± 0.01
Qsarbioconcentration	0.43 ± 0.00	0.44 ± 0.01	0.45 ± 0.04	0.62 ± 0.05	0.50 ± 0.01	0.52 ± 0.01	0.54 ± 0.03	0.47 ± 0.03	0.48 ± 0.02	0.52 ± 0.03
Qsarbiodegradation	0.42 ± 0.01	0.47 ± 0.03	0.38 ± 0.03	0.40 ± 0.03	0.37 ± 0.03	0.37 ± 0.03	0.38 ± 0.03	0.33 ± 0.02	0.33 ± 0.03	0.48 ± 0.04
Qualitative-bankruptcy	0.16 ± 0.02	0.03 ± 0.05	0.09 ± 0.11	0.01 ± 0.01	0.05 ± 0.10	0.05 ± 0.10	0.02 ± 0.06	0.02 ± 0.06	0.02 ± 0.06	0.07 ± 0.11
Ringnorm	0.37 ± 0.01	0.10 ± 0.01	0.28 ± 0.00	0.50 ± 0.02	0.12 ± 0.02	0.12 ± 0.02	0.13 ± 0.02	0.27 ± 0.02	0.27 ± 0.01	0.60 ± 0.01
Risk-factors-cervical-cancer	0.23 ± 0.01	0.31 ± 0.03	0.19 ± 0.04	0.23 ± 0.04	0.25 ± 0.01	0.26 ± 0.02	0.24 ± 0.03	0.20 ± 0.04	0.21 ± 0.04	0.19 ± 0.05
Robotnavigation	0.22 ± 0.01	0.44 ± 0.01	0.04 ± 0.01	0.24 ± 0.01	0.22 ± 0.01	0.21 ± 0.01	0.20 ± 0.01	0.23 ± 0.02	0.22 ± 0.02	0.35 ± 0.01
Sapfile	0.45 ± 0.01	0.46 ± 0.05	0.54 ± 0.07	0.60 ± 0.04	0.58 ± 0.05	0.58 ± 0.04	0.58 ± 0.05	0.54 ± 0.05	0.54 ± 0.07	0.62 ± 0.06
Sat	0.22 ± 0.00	0.26 ± 0.01	0.21 ± 0.01	0.18 ± 0.01	0.50 ± 0.00	0.49 ± 0.00	0.49 ± 0.00	0.17 ± 0.01	0.18 ± 0.01	0.37 ± 0.01
Satelite	0.22 ± 0.00	0.26 ± 0.01	0.21 ± 0.01	0.18 ± 0.01	0.50 ± 0.00	0.49 ± 0.00	0.49 ± 0.00	0.17 ± 0.00	0.18 ± 0.01	0.37 ± 0.01
Scadi	0.28 ± 0.01	0.20 ± 0.08	0.20 ± 0.08	0.21 ± 0.09	0.26 ± 0.05	0.23 ± 0.10	0.20 ± 0.12	0.22 ± 0.06	0.22 ± 0.07	0.34 ± 0.04
Schillingdata	0.29 ± 0.00	0.21 ± 0.01	0.34 ± 0.00	0.31 ± 0.01	0.36 ± 0.00	0.33 ± 0.02	0.29 ± 0.01	0.24 ± 0.03	0.29 ± 0.04	0.37 ± 0.01
Seeds	0.28 ± 0.03	0.21 ± 0.09	0.21 ± 0.10	0.19 ± 0.03	0.18 ± 0.10	0.18 ± 0.10	0.18 ± 0.10	0.14 ± 0.05	0.14 ± 0.05	0.33 ± 0.09
Segment	0.21 ± 0.00	0.23 ± 0.01	0.09 ± 0.02	0.09 ± 0.01	0.31 ± 0.01	0.31 ± 0.01	0.31 ± 0.01	0.10 ± 0.01	0.09 ± 0.01	0.32 ± 0.01
Seismic-bumps	0.25 ± 0.00	0.34 ± 0.02	0.25 ± 0.00	0.32 ± 0.02	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.25 ± 0.01	0.26 ± 0.01	0.27 ± 0.01
Semeion	0.29 ± 0.00	0.16 ± 0.02	0.21 ± 0.01	0.13 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.40 ± 0.00
Setapprocesst1	0.44 ± 0.03	0.61 ± 0.19	0.56 ± 0.14	0.57 ± 0.09	0.60 ± 0.07	0.55 ± 0.10	0.57 ± 0.09	0.54 ± 0.21	0.54 ± 0.21	0.45 ± 0.28
Setapprocesst10	0.47 ± 0.03	0.73 ± 0.10	0.62 ± 0.11	0.60 ± 0.15	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.63 ± 0.20	0.63 ± 0.17	0.61 ± 0.07
Setapprocesst11	0.47 ± 0.03	0.63 ± 0.19	0.55 ± 0.16	0.63 ± 0.12	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.57 ± 0.12	0.54 ± 0.13	0.65 ± 0.08
Setapprocesst2	0.47 ± 0.03	0.54 ± 0.14	0.43 ± 0.24	0.56 ± 0.23	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.56 ± 0.15	0.54 ± 0.16	0.37 ± 0.23
Setapprocesst3	0.47 ± 0.02	0.58 ± 0.11	0.60 ± 0.14	0.56 ± 0.12	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.55 ± 0.11	0.56 ± 0.08	0.68 ± 0.12
Setapprocesst4	0.46 ± 0.03	0.66 ± 0.21	0.64 ± 0.11	0.60 ± 0.16	0.53 ± 0.07	0.53 ± 0.07	0.53 ± 0.07	0.61 ± 0.17	0.62 ± 0.13	0.51 ± 0.11
Setapprocesst5	0.46 ± 0.03	0.71 ± 0.12	0.64 ± 0.16	0.61 ± 0.14	0.59 ± 0.06	0.60 ± 0.07	0.60 ± 0.07	0.58 ± 0.11	0.57 ± 0.10	0.53 ± 0.25
Setapprocesst6	0.46 ± 0.01	0.54 ± 0.15	0.55 ± 0.11	0.58 ± 0.12	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.55 ± 0.18	0.56 ± 0.16	0.61 ± 0.16
Setapprocesst7	0.47 ± 0.03	0.47 ± 0.17	0.56 ± 0.23	0.58 ± 0.13	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.46 ± 0.20	0.47 ± 0.20	0.68 ± 0.11
Setapprocesst8	0.46 ± 0.03	0.54 ± 0.13	0.49 ± 0.15	0.57 ± 0.20	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.52 ± 0.20	0.51 ± 0.21	0.71 ± 0.08
Setapprocesst9	0.47 ± 0.02	0.56 ± 0.14	0.52 ± 0.15	0.63 ± 0.07	0.58 ± 0.05	0.58 ± 0.05	0.58 ± 0.05	0.46 ± 0.21	0.52 ± 0.21	0.69 ± 0.12
Shillbiddingdataset	0.21 ± 0.00	0.17 ± 0.01	0.06 ± 0.01	0.09 ± 0.02	0.07 ± 0.01	0.06 ± 0.02	0.06 ± 0.02	0.05 ± 0.01	0.05 ± 0.02	0.16 ± 0.02
Shuttle-landing-control	0.19 ± 0.22	0.27 ± 0.32	0.14 ± 0.20	0.16 ± 0.23	0.07 ± 0.22	0.07 ± 0.22	0.07 ± 0.22	0.19 ± 0.31	0.19 ± 0.32	0.17 ± 0.37
Somervillehappinesssurvey2015	0.49 ± 0.03	0.52 ± 0.06	0.48 ± 0.10	0.56 ± 0.11	0.63 ± 0.08	0.64 ± 0.08	0.64 ± 0.09	0.59 ± 0.09	0.59 ± 0.10	0.57 ± 0.13
Sonar	0.46 ± 0.01	0.53 ± 0.07	0.52 ± 0.06	0.35 ± 0.10	0.58 ± 0.05	0.54 ± 0.08	0.51 ± 0.09	0.37 ± 0.15	0.37 ± 0.15	0.61 ± 0.09
Soybean	0.20 ± 0.00	0.08 ± 0.04	0.11 ± 0.02	0.11 ± 0.02	0.10 ± 0.03	0.09 ± 0.03	0.09 ± 0.03	0.09 ± 0.03	0.09 ± 0.02	0.28 ± 0.00
Spambase	0.42 ± 0.00	0.45 ± 0.02	0.26 ± 0.02	0.30 ± 0.02	0.26 ± 0.02	0.25 ± 0.02	0.26 ± 0.02	0.26 ± 0.02	0.29 ± 0.07	0.46 ± 0.02
Speaker-accent	0.31 ± 0.01	0.32 ± 0.04	0.30 ± 0.05	0.25 ± 0.06	0.37 ± 0.02	0.35 ± 0.02	0.35 ± 0.02	0.23 ± 0.03	0.25 ± 0.04	0.41 ± 0.02
Spect	0.45 ± 0.04	0.48 ± 0.18	0.48 ± 0.10	0.63 ± 0.11	0.59 ± 0.10	0.59 ± 0.11	0.60 ± 0.08	0.60 ± 0.14	0.60 ± 0.15	0.52 ± 0.10
Spectf	0.47 ± 0.02	0.46 ± 0.14	0.49 ± 0.12	0.54 ± 0.15	0.57 ± 0.06	0.57 ± 0.06	0.45 ± 0.19	0.48 ± 0.19	0.48 ± 0.14	0.53 ± 0.08
Statlog-australian-credit	0.46 ± 0.00	0.50 ± 0.03	0.51 ± 0.06	0.66 ± 0.02	0.57 ± 0.02	0.60 ± 0.05	0.60 ± 0.05	0.53 ± 0.03	0.54 ± 0.03	0.59 ± 0.05
Statlog-german-credit	0.45 ± 0.00	0.42 ± 0.04	0.47 ± 0.05	0.57 ± 0.03	0.49 ± 0.04	0.47 ± 0.04	0.47 ± 0.05	0.51 ± 0.03	0.50 ± 0.03	0.54 ± 0.02
Statlog-heart	0.41 ± 0.02	0.36 ± 0.07	0.45 ± 0.09	0.48 ± 0.11	0.41 ± 0.06	0.44 ± 0.10	0.44 ± 0.11	0.43 ± 0.06	0.41 ± 0.09	0.53 ± 0.06
Statlog-image	0.20 ± 0.00	0.23 ± 0.01	0.09 ± 0.02	0.09 ± 0.01	0.13 ± 0.01	0.12 ± 0.01	0.11 ± 0.01	0.09 ± 0.01	0.09 ± 0.02	0.32 ± 0.01

Table 8: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Statlog-landsat	0.22 ± 0.00	0.26 ± 0.01	0.21 ± 0.01	0.18 ± 0.02	0.19 ± 0.01	0.18 ± 0.01	0.18 ± 0.02	0.18 ± 0.01	0.18 ± 0.02	0.36 ± 0.01
Statlog-shuttle	0.17 ± 0.00	0.13 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.12 ± 0.00
Statlog-vehicle	0.36 ± 0.01	0.46 ± 0.01	0.33 ± 0.03	0.39 ± 0.03	0.34 ± 0.03	0.31 ± 0.02	0.31 ± 0.02	0.26 ± 0.02	0.27 ± 0.03	0.49 ± 0.03
Steel-plates	0.27 ± 0.00	0.32 ± 0.01	0.25 ± 0.02	0.28 ± 0.01	0.26 ± 0.02	0.26 ± 0.01	0.26 ± 0.02	0.25 ± 0.01	0.26 ± 0.02	0.38 ± 0.01
Synthetic-control	0.27 ± 0.01	0.13 ± 0.04	0.16 ± 0.03	0.10 ± 0.04	0.03 ± 0.04	0.02 ± 0.04	0.02 ± 0.04	0.04 ± 0.02	0.04 ± 0.02	0.38 ± 0.02
Teaching	0.45 ± 0.02	0.45 ± 0.03	0.46 ± 0.06	0.48 ± 0.08	0.53 ± 0.09	0.50 ± 0.08	0.50 ± 0.08	0.45 ± 0.04	0.46 ± 0.03	0.60 ± 0.06
Thoraricsurgery	0.35 ± 0.00	0.39 ± 0.07	0.37 ± 0.02	0.47 ± 0.06	0.39 ± 0.00	0.39 ± 0.01	0.39 ± 0.01	0.42 ± 0.05	0.41 ± 0.06	0.41 ± 0.02
Thyroid	0.27 ± 0.05	0.10 ± 0.09	0.19 ± 0.11	0.11 ± 0.09	0.40 ± 0.04	0.39 ± 0.04	0.39 ± 0.04	0.12 ± 0.08	0.12 ± 0.08	0.21 ± 0.14
Thyroid-train	0.20 ± 0.00	0.15 ± 0.02	0.03 ± 0.03	0.23 ± 0.02	0.18 ± 0.01	0.17 ± 0.01	0.16 ± 0.01	0.13 ± 0.03	0.15 ± 0.02	0.15 ± 0.02
Tic-tac-toe	0.46 ± 0.00	0.43 ± 0.01	0.23 ± 0.04	0.00 ± 0.00	0.09 ± 0.07	0.06 ± 0.07	0.02 ± 0.05	0.14 ± 0.03	0.13 ± 0.05	0.55 ± 0.02
Titanic	0.41 ± 0.01	0.42 ± 0.02	0.40 ± 0.02	0.39 ± 0.01	0.46 ± 0.02	0.46 ± 0.02	0.46 ± 0.02	0.40 ± 0.02	0.41 ± 0.02	0.47 ± 0.02
Trains	0.46 ± 0.08	0.40 ± 0.52	0.28 ± 0.46	0.42 ± 0.42	0.40 ± 0.52	0.30 ± 0.48	0.30 ± 0.48	0.28 ± 0.37	0.27 ± 0.37	1.00 ± 0.00
Transfusion	0.42 ± 0.01	0.43 ± 0.02	0.40 ± 0.02	0.53 ± 0.04	0.50 ± 0.03	0.52 ± 0.03	0.53 ± 0.03	0.39 ± 0.01	0.39 ± 0.01	0.49 ± 0.02
Trial	0.34 ± 0.01	0.30 ± 0.06	0.00 ± 0.00	0.02 ± 0.05	0.07 ± 0.06	0.06 ± 0.06	0.06 ± 0.06	0.06 ± 0.06	0.05 ± 0.06	0.00 ± 0.00
Turkiye-student-evaluation	0.40 ± 0.00	0.28 ± 0.02	0.01 ± 0.01	0.25 ± 0.02	0.17 ± 0.01	0.16 ± 0.01	0.16 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01
Unbalanced	0.12 ± 0.02	0.27 ± 0.06	0.12 ± 0.02	0.14 ± 0.06	0.12 ± 0.02	0.13 ± 0.02	0.13 ± 0.03	0.13 ± 0.02	0.13 ± 0.02	0.12 ± 0.02
Urbanlandcover	0.26 ± 0.01	0.21 ± 0.05	0.21 ± 0.04	0.22 ± 0.06	0.43 ± 0.01	0.43 ± 0.01	0.43 ± 0.01	0.19 ± 0.06	0.19 ± 0.06	0.33 ± 0.05
Userknowledgemodeling	0.36 ± 0.02	0.23 ± 0.05	0.15 ± 0.06	0.30 ± 0.09	0.33 ± 0.05	0.27 ± 0.04	0.27 ± 0.04	0.17 ± 0.05	0.15 ± 0.07	0.26 ± 0.08
Vehicle	0.35 ± 0.01	0.46 ± 0.02	0.33 ± 0.03	0.39 ± 0.03	0.59 ± 0.01	0.59 ± 0.01	0.59 ± 0.01	*0.26 ± 0.02*	0.27 ± 0.03	0.49 ± 0.03
Vertebral-column-2classes	0.40 ± 0.04	0.42 ± 0.09	0.37 ± 0.07	0.42 ± 0.06	0.37 ± 0.08	0.38 ± 0.09	0.38 ± 0.08	*0.32 ± 0.06*	*0.32 ± 0.06*	0.51 ± 0.08
Vertebral-column-3classes	0.36 ± 0.02	0.28 ± 0.04	0.31 ± 0.04	0.37 ± 0.04	0.32 ± 0.06	0.31 ± 0.07	0.32 ± 0.06	*0.26 ± 0.04*	*0.26 ± 0.04*	0.40 ± 0.06
Veteran	*0.43 ± 0.01*	0.46 ± 0.07	0.45 ± 0.06	0.62 ± 0.07	0.54 ± 0.01	0.54 ± 0.01	0.54 ± 0.01	0.54 ± 0.12	0.51 ± 0.12	0.51 ± 0.08
Vowel	0.24 ± 0.00	0.21 ± 0.01	0.17 ± 0.02	*0.03 ± 0.03*	0.16 ± 0.01	0.13 ± 0.01	0.11 ± 0.02	0.11 ± 0.02	0.10 ± 0.02	0.35 ± 0.01
Wall-following	0.23 ± 0.01	0.44 ± 0.01	*0.04 ± 0.02*	0.24 ± 0.01	0.23 ± 0.02	0.22 ± 0.02	0.21 ± 0.02	0.23 ± 0.01	0.23 ± 0.02	0.35 ± 0.01
Waveform-noise	0.41 ± 0.00	0.34 ± 0.01	0.39 ± 0.01	0.42 ± 0.01	*0.30 ± 0.02*	0.31 ± 0.01	0.32 ± 0.02	0.31 ± 0.02	0.32 ± 0.02	0.56 ± 0.02
Waveform	0.39 ± 0.00	0.33 ± 0.01	0.38 ± 0.01	0.39 ± 0.02	*0.30 ± 0.02*	0.31 ± 0.01	0.31 ± 0.02	*0.30 ± 0.01*	0.32 ± 0.01	0.56 ± 0.01
Wbc	0.18 ± 0.03	0.14 ± 0.06	0.23 ± 0.02	0.18 ± 0.06	0.13 ± 0.08	0.13 ± 0.08	*0.12 ± 0.09*	0.19 ± 0.04	0.18 ± 0.05	0.30 ± 0.05
Wdbc	0.28 ± 0.02	0.24 ± 0.11	0.24 ± 0.10	0.20 ± 0.05	0.61 ± 0.01	0.61 ± 0.01	0.61 ± 0.01	*0.16 ± 0.05*	0.17 ± 0.07	0.32 ± 0.05
Weathernominal	0.49 ± 0.16	0.44 ± 0.17	0.42 ± 0.37	0.52 ± 0.25	0.38 ± 0.41	0.48 ± 0.43	*0.31 ± 0.41*	*0.31 ± 0.40*	*0.31 ± 0.41*	0.61 ± 0.44
Weathernumeric	0.57 ± 0.15	0.49 ± 0.20	*0.27 ± 0.36*	*0.27 ± 0.34*	0.38 ± 0.41	0.54 ± 0.48	0.54 ± 0.48	0.31 ± 0.41	0.31 ± 0.41	0.61 ± 0.44
Website-phishingdata	0.35 ± 0.00	0.28 ± 0.02	*0.22 ± 0.02*	0.25 ± 0.02	0.31 ± 0.01	0.30 ± 0.01	0.29 ± 0.02	0.25 ± 0.02	0.25 ± 0.03	0.35 ± 0.02
Wholesalecustomersdata	0.29 ± 0.02	0.30 ± 0.05	0.29 ± 0.04	0.34 ± 0.07	0.57 ± 0.01	0.57 ± 0.01	0.57 ± 0.01	*0.25 ± 0.05*	0.26 ± 0.05	0.30 ± 0.06
Wifi-localization	0.26 ± 0.01	*0.09 ± 0.02*	0.11 ± 0.01	*0.09 ± 0.02*	0.32 ± 0.02	0.31 ± 0.02	0.31 ± 0.02	*0.09 ± 0.01*	*0.09 ± 0.01*	0.32 ± 0.03
Wilt	0.22 ± 0.00	0.27 ± 0.01	0.13 ± 0.02	0.23 ± 0.01	0.23 ± 0.00	0.23 ± 0.00	0.23 ± 0.00	*0.12 ± 0.03*	*0.12 ± 0.02*	0.24 ± 0.00
Wine-quality-red	*0.29 ± 0.01*	0.32 ± 0.01	0.33 ± 0.01	0.34 ± 0.01	0.35 ± 0.01	0.35 ± 0.01	0.35 ± 0.01	0.30 ± 0.01	0.30 ± 0.01	0.39 ± 0.01
Wine-quality-white	*0.29 ± 0.00*	0.32 ± 0.00	0.32 ± 0.01	0.31 ± 0.01	0.35 ± 0.01	0.35 ± 0.01	0.35 ± 0.01	*0.29 ± 0.00*	*0.29 ± 0.00*	0.39 ± 0.01
Wine	0.27 ± 0.02	0.09 ± 0.10	0.17 ± 0.11	0.15 ± 0.10	*0.04 ± 0.08*	0.06 ± 0.09	0.06 ± 0.09	0.09 ± 0.08	0.09 ± 0.08	0.38 ± 0.08
Yamilnaduelectricty	*0.21 ± 0.00*	0.22 ± 0.00	0.25 ± 0.00	0.28 ± 0.00	0.31 ± 0.00	0.31 ± 0.00	0.31 ± 0.00	0.22 ± 0.00	0.22 ± 0.00	0.30 ± 0.00
Yeast	0.26 ± 0.00	*0.24 ± 0.01*	0.27 ± 0.01	0.31 ± 0.01	0.28 ± 0.01	0.28 ± 0.01	0.28 ± 0.02	*0.24 ± 0.01*	*0.24 ± 0.01*	0.35 ± 0.01
Youtobe-kabita-preprocessing	*0.33 ± 0.00*	0.37 ± 0.00	0.35 ± 0.01	0.40 ± 0.00	0.42 ± 0.01	0.43 ± 0.01	0.43 ± 0.01	*0.33 ± 0.00*	*0.33 ± 0.00*	0.44 ± 0.01
Youtobe-nisha-preprocessing	*0.33 ± 0.00*	0.36 ± 0.00	0.35 ± 0.01	0.39 ± 0.01	0.41 ± 0.01	0.42 ± 0.01	0.42 ± 0.01	*0.33 ± 0.00*	*0.33 ± 0.00*	0.44 ± 0.00
Z-alizadehsani	0.43 ± 0.01	0.39 ± 0.07	0.43 ± 0.09	0.46 ± 0.07	0.54 ± 0.01	0.54 ± 0.01	0.54 ± 0.01	*0.37 ± 0.09*	0.39 ± 0.07	0.54 ± 0.05
Zoo	0.23 ± 0.02	*0.06 ± 0.09*	0.10 ± 0.10	*0.06 ± 0.07*	0.08 ± 0.11	0.08 ± 0.11	0.08 ± 0.11	0.07 ± 0.07	0.07 ± 0.07	0.27 ± 0.06
Average rmse (rank)	0.3321 (5)	0.3377 (6)	0.2832 (2)	0.3247 (4)	0.3630 (9)	0.3562 (8)	0.3518 (7)	0.2817 (1)	0.2841 (3)	0.3928 (10)
Average rmse std (rank)	0.0152 (1)	0.0496 (4)	0.0531 (7)	0.0570 (9)	0.0442 (2)	0.0478 (3)	0.0497 (5)	0.0554 (8)	0.0573 (10)	0.0513 (6)

Table 9: The detailed experimental results on mean absolute error and standard deviation.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Abalone	0.34 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.28 ± 0.02	0.22 ± 0.02	0.22 ± 0.02	0.22 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.26 ± 0.01
Absenteeism-at-work	0.31 ± 0.01	0.16 ± 0.03	0.00 ± 0.01	0.16 ± 0.03	0.15 ± 0.03	0.15 ± 0.03	0.15 ± 0.03	0.04 ± 0.01	0.04 ± 0.02	0.05 ± 0.02
Acute-inflammation	0.23 ± 0.02	0.10 ± 0.03	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.21 ± 0.08
Acute-nephritis	0.21 ± 0.01	0.05 ± 0.05	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.08 ± 0.08
Adult	0.34 ± 0.00	0.20 ± 0.02	0.22 ± 0.03	0.22 ± 0.04	0.17 ± 0.04	0.17 ± 0.04	0.17 ± 0.04	0.21 ± 0.03	0.21 ± 0.03	0.20 ± 0.01
Aggregation	0.04 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00	0.01 ± 0.01	0.12 ± 0.01
Algerianforest	0.30 ± 0.02	0.06 ± 0.05	0.05 ± 0.04	0.12 ± 0.07	0.41 ± 0.02	0.38 ± 0.04	0.38 ± 0.04	0.04 ± 0.03	0.04 ± 0.04	0.03 ± 0.05
Annealing	0.15 ± 0.00	0.12 ± 0.01	0.02 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.07 ± 0.00
Arrhythmia	0.10 ± 0.00	0.06 ± 0.01	0.06 ± 0.01	0.07 ± 0.00	0.06 ± 0.00	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.07 ± 0.02	0.06 ± 0.00
Au1-1000	0.38 ± 0.00	0.36 ± 0.01	0.30 ± 0.03	0.32 ± 0.04	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.01	0.31 ± 0.04	0.29 ± 0.04	0.26 ± 0.00
Au4-2500	0.36 ± 0.00	0.32 ± 0.01	0.20 ± 0.03	0.24 ± 0.03	0.26 ± 0.02	0.26 ± 0.02	0.26 ± 0.02	0.24 ± 0.03	0.24 ± 0.03	0.31 ± 0.03
Au6-1000	0.21 ± 0.00	0.21 ± 0.00	0.20 ± 0.01	0.22 ± 0.01	0.19 ± 0.00	0.19 ± 0.00	0.19 ± 0.00	0.21 ± 0.01	0.22 ± 0.01	0.19 ± 0.01
Au6-250-drift-au6-cd1-500	0.21 ± 0.00	0.21 ± 0.00	0.20 ± 0.01	0.21 ± 0.01	0.19 ± 0.00	0.19 ± 0.00	0.19 ± 0.00	0.21 ± 0.00	0.21 ± 0.01	0.19 ± 0.01
Au6-cd1-400	0.21 ± 0.00	0.20 ± 0.01	0.16 ± 0.02	0.21 ± 0.02	0.18 ± 0.00	0.18 ± 0.00	0.18 ± 0.00	0.20 ± 0.01	0.20 ± 0.01	0.17 ± 0.02
Au7-300-drift-au7-cpd1-800	0.31 ± 0.00	0.29 ± 0.00	0.26 ± 0.01	0.27 ± 0.02	0.29 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.28 ± 0.01	0.28 ± 0.01	0.30 ± 0.01
Au7-700	0.26 ± 0.00	0.24 ± 0.01	0.20 ± 0.01	0.26 ± 0.02	0.25 ± 0.01	0.25 ± 0.01	0.25 ± 0.01	0.24 ± 0.01	0.24 ± 0.01	0.22 ± 0.02
Au7-cpd1-500	0.30 ± 0.00	0.28 ± 0.01	0.21 ± 0.02	0.25 ± 0.02	0.25 ± 0.00	0.25 ± 0.01	0.25 ± 0.01	0.25 ± 0.02	0.25 ± 0.02	0.24 ± 0.02
Audiology-std	0.09 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.06 ± 0.00
Audit-risk	0.28 ± 0.01	0.05 ± 0.02	0.00 ± 0.00	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.04 ± 0.02	0.04 ± 0.02	0.00 ± 0.00
Autism-adolescent-data	0.37 ± 0.03	0.04 ± 0.03	0.00 ± 0.00	0.10 ± 0.06	0.04 ± 0.05	0.05 ± 0.05	0.04 ± 0.05	0.10 ± 0.09	0.10 ± 0.09	0.00 ± 0.00
Autism-adult-data	0.29 ± 0.01	0.03 ± 0.02	0.00 ± 0.00	0.05 ± 0.04	0.01 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Autism-child-data	0.40 ± 0.01	0.04 ± 0.02	0.00 ± 0.00	0.12 ± 0.04	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00
Autos	0.17 ± 0.01	0.13 ± 0.03	0.06 ± 0.02	0.07 ± 0.03	0.19 ± 0.01	0.18 ± 0.01	0.18 ± 0.01	0.07 ± 0.02	0.07 ± 0.01	0.11 ± 0.03
Avila	0.07 ± 0.00	0.13 ± 0.00	0.01 ± 0.00	0.04 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.05 ± 0.00
Balance-scale	0.28 ± 0.01	0.21 ± 0.01	0.19 ± 0.02	0.13 ± 0.03	0.06 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.29 ± 0.03
Balloons	0.41 ± 0.12	0.50 ± 0.03	0.46 ± 0.19	0.34 ± 0.12	0.30 ± 0.35	0.20 ± 0.35	0.20 ± 0.35	0.28 ± 0.23	0.27 ± 0.24	0.55 ± 0.37
Bank	0.20 ± 0.00	0.18 ± 0.01	0.14 ± 0.01	0.14 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.13 ± 0.01	0.13 ± 0.02	0.11 ± 0.01
Blood	0.35 ± 0.01	0.29 ± 0.02	0.30 ± 0.01	0.32 ± 0.04	0.22 ± 0.02	0.21 ± 0.02	0.22 ± 0.01	0.30 ± 0.02	0.30 ± 0.02	0.24 ± 0.02
Breast-cancer-wisc-diag	0.25 ± 0.02	0.07 ± 0.04	0.07 ± 0.04	0.04 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.02 ± 0.01	0.04 ± 0.02	0.04 ± 0.02	0.11 ± 0.04
Breast-cancer-wisc-prog	0.36 ± 0.01	0.34 ± 0.09	0.29 ± 0.06	0.27 ± 0.09	0.21 ± 0.06	0.22 ± 0.09	0.22 ± 0.10	0.27 ± 0.09	0.27 ± 0.11	0.30 ± 0.08
Breast-cancer-wisc	0.14 ± 0.02	0.04 ± 0.02	0.07 ± 0.03	0.05 ± 0.03	0.03 ± 0.02	0.03 ± 0.02	0.03 ± 0.02	0.06 ± 0.03	0.05 ± 0.02	0.07 ± 0.05
Breast-cancer	0.39 ± 0.01	0.32 ± 0.04	0.37 ± 0.03	0.30 ± 0.08	0.26 ± 0.05	0.27 ± 0.05	0.28 ± 0.05	0.32 ± 0.06	0.33 ± 0.04	0.30 ± 0.05
Breast-tissue	0.19 ± 0.02	0.11 ± 0.04	0.11 ± 0.05	0.11 ± 0.04	0.14 ± 0.05	0.11 ± 0.06	0.10 ± 0.06	0.14 ± 0.02	0.14 ± 0.03	0.15 ± 0.03
Bupa	0.45 ± 0.01	0.46 ± 0.03	0.40 ± 0.08	0.38 ± 0.05	0.41 ± 0.02	0.41 ± 0.02	0.41 ± 0.02	0.37 ± 0.04	0.36 ± 0.04	0.46 ± 0.08
Caesarian	0.43 ± 0.05	0.41 ± 0.10	0.46 ± 0.06	0.47 ± 0.14	0.45 ± 0.06	0.38 ± 0.12	0.41 ± 0.17	0.44 ± 0.18	0.44 ± 0.17	0.51 ± 0.18
Car	0.22 ± 0.00	0.12 ± 0.01	0.01 ± 0.00	0.02 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.00	0.04 ± 0.01	0.04 ± 0.01	0.15 ± 0.00
Cardiotocography-10classes	0.14 ± 0.00	0.06 ± 0.00	0.04 ± 0.00	0.04 ± 0.01	0.04 ± 0.00	0.04 ± 0.01	0.03 ± 0.00	0.04 ± 0.01	0.04 ± 0.00	0.10 ± 0.00
Cardiotocography-3classes	0.22 ± 0.00	0.12 ± 0.02	0.06 ± 0.02	0.05 ± 0.01	0.06 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.12 ± 0.02
Cervical-cancer	0.34 ± 0.03	0.10 ± 0.07	0.16 ± 0.12	0.12 ± 0.11	0.29 ± 0.03	0.26 ± 0.07	0.26 ± 0.07	0.08 ± 0.08	0.07 ± 0.08	0.22 ± 0.09
Chemicalcomposionofceramic	0.24 ± 0.02	0.00 ± 0.00	0.01 ± 0.04	0.01 ± 0.00	0.54 ± 0.02	0.54 ± 0.02	0.54 ± 0.02	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Chess-krvk	0.10 ± 0.00	0.09 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.07 ± 0.00	0.07 ± 0.00	0.08 ± 0.00
Chess-krvkp	0.47 ± 0.00	0.21 ± 0.02	0.01 ± 0.00	0.10 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.34 ± 0.01
Congressional-voting	0.47 ± 0.00	0.45 ± 0.04	0.46 ± 0.03	0.45 ± 0.05	0.38 ± 0.04	0.39 ± 0.05	0.39 ± 0.06	0.43 ± 0.03	0.43 ± 0.03	0.37 ± 0.03
Conn-bench-sonar-mines-rocks	0.45 ± 0.01	0.31 ± 0.09	0.29 ± 0.07	0.14 ± 0.07	0.16 ± 0.07	0.13 ± 0.08	0.12 ± 0.08	0.20 ± 0.09	0.18 ± 0.09	0.37 ± 0.10

Table 9: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Conn-bench-vowel-deterding	0.11 ± 0.00	0.08 ± 0.01	0.04 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.04 ± 0.01	0.04 ± 0.01	0.12 ± 0.01
Connect-4	0.38 ± 0.00	0.38 ± 0.01	0.24 ± 0.02	0.29 ± 0.03	0.26 ± 0.00	0.24 ± 0.01	0.23 ± 0.02	0.27 ± 0.02	0.26 ± 0.02	0.26 ± 0.00
Connectionist	0.45 ± 0.01	0.32 ± 0.11	0.27 ± 0.07	0.13 ± 0.06	0.39 ± 0.09	0.30 ± 0.12	0.27 ± 0.12	0.17 ± 0.08	0.17 ± 0.08	0.37 ± 0.07
Contrac	0.42 ± 0.00	0.37 ± 0.02	0.34 ± 0.02	0.38 ± 0.02	0.30 ± 0.01	0.30 ± 0.02	0.31 ± 0.03	0.35 ± 0.01	0.34 ± 0.01	0.35 ± 0.01
Covid-19	0.32 ± 0.08	0.29 ± 0.17	0.24 ± 0.24	0.33 ± 0.23	0.27 ± 0.26	0.30 ± 0.29	0.23 ± 0.27	0.23 ± 0.23	0.22 ± 0.23	0.13 ± 0.23
Credit-approval	0.41 ± 0.01	0.23 ± 0.03	0.19 ± 0.03	0.18 ± 0.06	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.05	0.16 ± 0.03	0.17 ± 0.04	0.14 ± 0.04
Crowdsource	0.12 ± 0.00	0.06 ± 0.01	0.04 ± 0.00	0.02 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	0.07 ± 0.01
Crx	0.41 ± 0.01	0.22 ± 0.03	0.19 ± 0.03	0.19 ± 0.05	0.44 ± 0.04	0.43 ± 0.04	0.42 ± 0.05	0.18 ± 0.04	0.17 ± 0.02	0.14 ± 0.04
Cryother	0.32 ± 0.05	0.18 ± 0.11	0.11 ± 0.07	0.11 ± 0.08	0.18 ± 0.13	0.17 ± 0.13	0.17 ± 0.13	0.14 ± 0.08	0.17 ± 0.09	0.19 ± 0.11
Cylinder-bands	0.45 ± 0.01	0.33 ± 0.06	0.29 ± 0.06	0.31 ± 0.07	0.24 ± 0.04	0.21 ± 0.04	0.19 ± 0.03	0.27 ± 0.05	0.26 ± 0.05	0.33 ± 0.04
Dbworld-bodies	0.49 ± 0.01	0.24 ± 0.15	0.21 ± 0.16	0.41 ± 0.10	0.45 ± 0.06	0.45 ± 0.06	0.36 ± 0.13	/	/	0.14 ± 0.13
Dbworld-bodies-stemmed	0.48 ± 0.01	0.23 ± 0.13	0.15 ± 0.13	0.36 ± 0.14	0.45 ± 0.06	0.41 ± 0.09	0.26 ± 0.13	/	/	0.26 ± 0.20
Dbworld-subjects	0.49 ± 0.01	0.17 ± 0.10	0.30 ± 0.10	0.25 ± 0.12	0.45 ± 0.06	0.45 ± 0.06	0.45 ± 0.06	0.13 ± 0.12	0.13 ± 0.14	0.37 ± 0.12
Dbworld-subjects-stemmed	0.49 ± 0.01	0.17 ± 0.11	0.27 ± 0.14	0.19 ± 0.09	0.45 ± 0.06	0.45 ± 0.06	0.45 ± 0.06	0.14 ± 0.12	0.14 ± 0.11	0.39 ± 0.10
Dermatology	0.19 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.17 ± 0.01
Diabetes	0.39 ± 0.02	0.28 ± 0.03	0.32 ± 0.05	0.30 ± 0.05	0.35 ± 0.00	0.35 ± 0.00	0.35 ± 0.00	0.30 ± 0.02	0.29 ± 0.02	0.29 ± 0.05
Diabetic	0.46 ± 0.01	0.43 ± 0.02	0.38 ± 0.03	0.39 ± 0.05	0.42 ± 0.05	0.41 ± 0.06	0.40 ± 0.06	0.33 ± 0.03	0.33 ± 0.03	0.47 ± 0.06
Divorce	0.10 ± 0.03	0.02 ± 0.03	0.06 ± 0.06	0.03 ± 0.03	0.02 ± 0.03	0.02 ± 0.03	0.02 ± 0.03	0.02 ± 0.03	0.02 ± 0.03	0.05 ± 0.05
Dota2train	0.50 ± 0.00	0.47 ± 0.02	0.47 ± 0.04	0.45 ± 0.03	0.47 ± 0.00	0.46 ± 0.04	0.47 ± 0.03	0.50 ± 0.03	0.48 ± 0.02	0.45 ± 0.04
Dow-jones-index	0.45 ± 0.02	0.50 ± 0.03	0.29 ± 0.04	0.45 ± 0.06	0.48 ± 0.00	0.48 ± 0.00	0.48 ± 0.00	0.46 ± 0.03	0.45 ± 0.03	0.44 ± 0.03
Dry-bean-dataset	0.06 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.11 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.03 ± 0.00	0.03 ± 0.00	0.09 ± 0.01
Early-stage-diabetes-data-upload	0.37 ± 0.01	0.15 ± 0.03	0.05 ± 0.03	0.02 ± 0.02	0.06 ± 0.04	0.05 ± 0.04	0.05 ± 0.04	0.04 ± 0.02	0.04 ± 0.03	0.18 ± 0.04
Echocardiogram	0.40 ± 0.03	0.22 ± 0.11	0.24 ± 0.10	0.26 ± 0.13	0.18 ± 0.08	0.19 ± 0.11	0.19 ± 0.11	0.24 ± 0.12	0.23 ± 0.12	0.15 ± 0.07
Ecoli	0.14 ± 0.00	0.04 ± 0.01	0.05 ± 0.02	0.05 ± 0.02	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.08 ± 0.02
Eegeyesate	0.45 ± 0.00	0.53 ± 0.02	0.17 ± 0.01	0.16 ± 0.02	0.45 ± 0.00	0.45 ± 0.00	0.45 ± 0.00	0.49 ± 0.01	0.48 ± 0.01	0.37 ± 0.01
Electrical	0.30 ± 0.03	0.06 ± 0.03	0.00 ± 0.00	0.09 ± 0.07	0.17 ± 0.08	0.17 ± 0.07	0.16 ± 0.06	0.02 ± 0.02	0.02 ± 0.01	0.00 ± 0.01
Energy-y1	0.25 ± 0.00	0.13 ± 0.01	0.02 ± 0.01	0.16 ± 0.03	0.08 ± 0.01	0.07 ± 0.01	0.07 ± 0.02	0.09 ± 0.01	0.09 ± 0.01	0.10 ± 0.01
Energy-y2	0.23 ± 0.00	0.11 ± 0.01	0.09 ± 0.01	0.16 ± 0.02	0.07 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.08 ± 0.01	0.08 ± 0.02	0.08 ± 0.02
Extentionofz-alizadehsani	0.38 ± 0.01	0.08 ± 0.04	0.01 ± 0.01	0.10 ± 0.05	0.29 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.02 ± 0.02	0.02 ± 0.02	0.14 ± 0.05
Fertility	0.21 ± 0.03	0.20 ± 0.03	0.22 ± 0.04	0.18 ± 0.11	0.12 ± 0.04	0.11 ± 0.06	0.11 ± 0.07	0.13 ± 0.08	0.13 ± 0.10	0.12 ± 0.04
First-order	0.17 ± 0.00	0.68 ± 0.04	0.00 ± 0.00	0.01 ± 0.00	0.10 ± 0.00	0.05 ± 0.01	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.09 ± 0.02
Flags	0.18 ± 0.00	0.13 ± 0.01	0.11 ± 0.02	0.14 ± 0.02	0.12 ± 0.02	0.11 ± 0.03	0.12 ± 0.03	0.14 ± 0.03	0.13 ± 0.03	0.11 ± 0.02
Foresttypes	0.23 ± 0.01	0.03 ± 0.02	0.03 ± 0.03	0.03 ± 0.02	0.35 ± 0.01	0.34 ± 0.02	0.34 ± 0.02	0.03 ± 0.02	0.02 ± 0.02	0.09 ± 0.04
Garments-worker-productivity	0.29 ± 0.00	0.29 ± 0.01	0.22 ± 0.02	0.30 ± 0.01	0.26 ± 0.01	0.26 ± 0.01	0.25 ± 0.01	0.26 ± 0.01	0.26 ± 0.01	0.27 ± 0.01
Gender-name-dataset	0.47 ± 0.00	0.42 ± 0.04	0.48 ± 0.00	0.41 ± 0.01	0.36 ± 0.01	0.36 ± 0.02	0.36 ± 0.02	0.48 ± 0.01	0.48 ± 0.01	0.39 ± 0.01
Gesture-a1-raw	0.12 ± 0.00	0.11 ± 0.01	0.04 ± 0.01	0.02 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.04 ± 0.01	0.04 ± 0.00	0.01 ± 0.01
Gesture-a1-va3	0.21 ± 0.01	0.16 ± 0.01	0.14 ± 0.02	0.10 ± 0.01	0.24 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.14 ± 0.01	0.14 ± 0.01	0.15 ± 0.01
Gesture-a2-raw	0.13 ± 0.00	0.13 ± 0.01	0.05 ± 0.01	0.02 ± 0.01	0.24 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.05 ± 0.01	0.05 ± 0.01	0.02 ± 0.01
Gesture-a2-va3	0.23 ± 0.01	0.23 ± 0.03	0.18 ± 0.01	0.13 ± 0.02	0.24 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.18 ± 0.01	0.18 ± 0.02	0.18 ± 0.02
Gesture-a3-raw	0.10 ± 0.00	0.16 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.05 ± 0.01	0.05 ± 0.01	0.02 ± 0.00
Gesture-a3-va3	0.24 ± 0.00	0.20 ± 0.01	0.10 ± 0.01	0.04 ± 0.01	0.26 ± 0.00	0.26 ± 0.00	0.26 ± 0.00	0.17 ± 0.01	0.16 ± 0.01	0.20 ± 0.01
Gesture-b1-raw	0.10 ± 0.01	0.18 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.20 ± 0.00	0.20 ± 0.00	0.20 ± 0.00	0.05 ± 0.01	0.04 ± 0.01	0.03 ± 0.01
Gesture-b1-va3	0.24 ± 0.01	0.26 ± 0.03	0.13 ± 0.02	0.04 ± 0.01	0.25 ± 0.00	0.25 ± 0.00	0.25 ± 0.00	0.18 ± 0.01	0.17 ± 0.02	0.23 ± 0.02
Gesture-b3-raw	0.15 ± 0.01	0.18 ± 0.01	0.04 ± 0.01	0.02 ± 0.00	0.27 ± 0.00	0.27 ± 0.00	0.27 ± 0.00	0.06 ± 0.01	0.06 ± 0.01	0.02 ± 0.01

Table 9: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Gesture-b3-va3	0.28 ± 0.00	0.23 ± 0.01	0.18 ± 0.01	0.09 ± 0.01	0.27 ± 0.00	0.27 ± 0.00	0.27 ± 0.00	0.19 ± 0.01	0.19 ± 0.01	0.24 ± 0.02
Gesture-c1-raw	0.15 ± 0.01	0.15 ± 0.01	0.05 ± 0.01	0.02 ± 0.01	0.30 ± 0.00	0.30 ± 0.00	0.30 ± 0.00	0.06 ± 0.01	0.06 ± 0.01	0.02 ± 0.01
Gesture-c1-va3	0.26 ± 0.01	0.20 ± 0.01	0.17 ± 0.01	0.12 ± 0.01	0.30 ± 0.00	0.30 ± 0.00	0.30 ± 0.00	0.16 ± 0.01	0.16 ± 0.01	0.23 ± 0.02
Gesture-c3-raw	0.14 ± 0.01	0.16 ± 0.02	0.05 ± 0.01	0.03 ± 0.01	0.29 ± 0.00	0.29 ± 0.00	0.29 ± 0.00	0.07 ± 0.01	0.07 ± 0.01	0.02 ± 0.01
Gesture-c3-va3	0.27 ± 0.01	0.22 ± 0.01	0.19 ± 0.02	0.15 ± 0.02	0.29 ± 0.00	0.29 ± 0.00	0.29 ± 0.00	0.19 ± 0.01	0.20 ± 0.02	0.23 ± 0.01
Glass	0.18 ± 0.01	0.18 ± 0.02	0.12 ± 0.02	0.10 ± 0.03	0.09 ± 0.02	0.09 ± 0.03	0.10 ± 0.03	0.13 ± 0.02	0.12 ± 0.01	0.14 ± 0.03
Go-track-tracks	0.34 ± 0.04	0.19 ± 0.05	0.14 ± 0.07	0.10 ± 0.08	0.28 ± 0.13	0.28 ± 0.12	0.28 ± 0.12	0.19 ± 0.06	0.18 ± 0.05	0.24 ± 0.14
Haberman-survival	0.37 ± 0.01	0.33 ± 0.03	0.36 ± 0.03	0.34 ± 0.07	0.27 ± 0.05	0.29 ± 0.03	0.29 ± 0.03	0.35 ± 0.03	0.35 ± 0.03	0.27 ± 0.03
Hayes-roth	0.27 ± 0.02	0.29 ± 0.02	0.13 ± 0.03	0.09 ± 0.04	0.09 ± 0.05	0.13 ± 0.04	0.13 ± 0.04	0.22 ± 0.07	0.21 ± 0.07	0.37 ± 0.04
Hcc-data	0.45 ± 0.01	0.33 ± 0.06	0.43 ± 0.06	0.37 ± 0.08	0.38 ± 0.02	0.38 ± 0.02	0.38 ± 0.02	0.36 ± 0.12	0.36 ± 0.10	0.31 ± 0.11
Hcvdat	0.08 ± 0.00	0.04 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	0.05 ± 0.01
Heart-cleveland	0.24 ± 0.01	0.18 ± 0.03	0.21 ± 0.02	0.18 ± 0.03	0.17 ± 0.03	0.16 ± 0.04	0.17 ± 0.04	0.19 ± 0.02	0.18 ± 0.02	0.19 ± 0.03
Heart-hungarian	0.36 ± 0.02	0.19 ± 0.07	0.27 ± 0.05	0.23 ± 0.08	0.18 ± 0.08	0.18 ± 0.06	0.17 ± 0.06	0.23 ± 0.08	0.23 ± 0.07	0.21 ± 0.09
Heart-switzerland	0.28 ± 0.01	0.26 ± 0.02	0.26 ± 0.04	0.27 ± 0.05	0.24 ± 0.03	0.24 ± 0.02	0.23 ± 0.03	0.24 ± 0.05	0.23 ± 0.04	0.27 ± 0.05
Heart-va	0.30 ± 0.00	0.28 ± 0.02	0.28 ± 0.02	0.26 ± 0.03	0.26 ± 0.04	0.27 ± 0.03	0.28 ± 0.04	0.29 ± 0.02	0.29 ± 0.03	0.29 ± 0.04
Heart-failure-clinical-records-dataset	0.41 ± 0.01	0.25 ± 0.04	0.24 ± 0.05	0.35 ± 0.07	0.32 ± 0.02	0.32 ± 0.02	0.32 ± 0.02	0.27 ± 0.05	0.24 ± 0.08	0.14 ± 0.06
Hepatitis	0.30 ± 0.02	0.18 ± 0.08	0.24 ± 0.07	0.18 ± 0.05	0.17 ± 0.06	0.16 ± 0.09	0.16 ± 0.09	0.20 ± 0.09	0.22 ± 0.11	0.26 ± 0.07
Hill-valley	0.50 ± 0.01	0.51 ± 0.02	0.50 ± 0.00	0.48 ± 0.03	0.48 ± 0.07	0.48 ± 0.07	0.47 ± 0.07	0.46 ± 0.02	0.45 ± 0.02	0.52 ± 0.05
Hiv1625data	0.18 ± 0.01	0.07 ± 0.02	0.13 ± 0.02	0.11 ± 0.01	0.08 ± 0.02	0.07 ± 0.02	0.06 ± 0.02	0.05 ± 0.02	0.07 ± 0.02	0.21 ± 0.02
Hiv746data	0.25 ± 0.02	0.11 ± 0.03	0.25 ± 0.03	0.17 ± 0.03	0.10 ± 0.04	0.09 ± 0.03	0.09 ± 0.03	0.08 ± 0.02	0.08 ± 0.03	0.19 ± 0.04
Horse-colic	0.41 ± 0.01	0.27 ± 0.06	0.16 ± 0.04	0.24 ± 0.07	0.16 ± 0.05	0.15 ± 0.05	0.15 ± 0.06	0.23 ± 0.05	0.22 ± 0.07	0.18 ± 0.05
Htru	0.06 ± 0.00	0.06 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.02 ± 0.00
Hypothyroid	0.07 ± 0.00	0.04 ± 0.00	0.00 ± 0.00	0.04 ± 0.01	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.02	0.04 ± 0.03	0.02 ± 0.00
Ibeacon-rssi-labeled	0.02 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Ilpd-indian-liver	0.39 ± 0.01	0.44 ± 0.05	0.39 ± 0.03	0.35 ± 0.06	0.29 ± 0.01	0.29 ± 0.02	0.29 ± 0.03	0.34 ± 0.03	0.34 ± 0.03	0.34 ± 0.03
Image-segmentation	0.15 ± 0.01	0.06 ± 0.02	0.04 ± 0.02	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.12 ± 0.03
Immunotherapy	0.30 ± 0.02	0.31 ± 0.06	0.23 ± 0.08	0.30 ± 0.16	0.21 ± 0.04	0.21 ± 0.04	0.21 ± 0.04	0.19 ± 0.08	0.20 ± 0.08	0.16 ± 0.09
Impensdata	0.21 ± 0.01	0.13 ± 0.02	0.27 ± 0.00	0.16 ± 0.02	0.16 ± 0.00	0.16 ± 0.00	0.16 ± 0.00	0.10 ± 0.02	0.11 ± 0.03	0.16 ± 0.02
In-vehicle-coupon-recommendation	0.48 ± 0.00	0.41 ± 0.01	0.38 ± 0.01	0.36 ± 0.01	0.31 ± 0.01	0.31 ± 0.01	0.30 ± 0.01	0.30 ± 0.01	0.29 ± 0.01	0.39 ± 0.01
Indian	0.38 ± 0.01	0.44 ± 0.05	0.33 ± 0.04	0.36 ± 0.07	0.28 ± 0.02	0.28 ± 0.01	0.28 ± 0.01	0.35 ± 0.03	0.34 ± 0.03	0.34 ± 0.03
Ionosphere	0.24 ± 0.02	0.17 ± 0.05	0.09 ± 0.03	0.14 ± 0.05	0.06 ± 0.02	0.05 ± 0.03	0.04 ± 0.03	0.10 ± 0.02	0.10 ± 0.03	0.19 ± 0.07
Iris	0.17 ± 0.02	0.04 ± 0.03	0.03 ± 0.03	0.04 ± 0.04	0.02 ± 0.02	0.03 ± 0.03	0.02 ± 0.02	0.03 ± 0.03	0.03 ± 0.02	0.05 ± 0.04
Jain	0.08 ± 0.03	0.07 ± 0.03	0.01 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.07 ± 0.02	0.07 ± 0.02	0.05 ± 0.03
Jsbach-chorals-harmony	0.17 ± 0.00	0.05 ± 0.01	0.05 ± 0.01	0.03 ± 0.00	0.04 ± 0.01	0.03 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.04 ± 0.01
Knowledge	0.27 ± 0.01	0.13 ± 0.04	0.04 ± 0.03	0.13 ± 0.04	0.08 ± 0.05	0.06 ± 0.04	0.04 ± 0.04	0.05 ± 0.02	0.05 ± 0.03	0.09 ± 0.05
Lasvegastripadvisorreviews	0.24 ± 0.00	0.24 ± 0.01	0.24 ± 0.01	0.24 ± 0.01	0.25 ± 0.01	0.25 ± 0.01	0.25 ± 0.01	0.24 ± 0.01	0.24 ± 0.01	0.24 ± 0.01
Leaf	0.05 ± 0.00	0.02 ± 0.00	0.03 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.05 ± 0.00
Led-display	0.15 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.06 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.08 ± 0.00	0.07 ± 0.00	0.16 ± 0.00
Lenses	0.28 ± 0.07	0.37 ± 0.09	0.15 ± 0.19	0.13 ± 0.11	0.14 ± 0.20	0.11 ± 0.15	0.11 ± 0.15	0.19 ± 0.15	0.20 ± 0.16	0.22 ± 0.20
Letter	0.07 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.06 ± 0.00
Libras	0.10 ± 0.00	0.05 ± 0.01	0.04 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.11 ± 0.01
Low-res-spect	0.11 ± 0.00	0.04 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.06 ± 0.01
Lung-cancer	0.41 ± 0.02	0.25 ± 0.11	0.39 ± 0.16	0.34 ± 0.22	0.31 ± 0.15	0.27 ± 0.14	0.32 ± 0.17	0.35 ± 0.14	0.35 ± 0.15	0.35 ± 0.12
Lymphography	0.23 ± 0.00	0.10 ± 0.03	0.14 ± 0.04	0.13 ± 0.06	0.07 ± 0.05	0.07 ± 0.05	0.08 ± 0.04	0.10 ± 0.04	0.10 ± 0.03	0.12 ± 0.04

Table 9: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Magic	0.34 ± 0.01	0.28 ± 0.02	0.22 ± 0.02	0.20 ± 0.02	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.01	0.19 ± 0.02	0.20 ± 0.02	0.28 ± 0.04
Mammographic	0.36 ± 0.01	0.24 ± 0.03	0.25 ± 0.02	0.26 ± 0.05	0.17 ± 0.02	0.18 ± 0.03	0.17 ± 0.03	0.25 ± 0.02	0.26 ± 0.03	0.18 ± 0.01
Miniboone	0.30 ± 0.00	0.71 ± 0.01	0.14 ± 0.02	0.17 ± 0.02	0.15 ± 0.02	0.14 ± 0.02	0.14 ± 0.02	0.21 ± 0.03	0.21 ± 0.03	0.18 ± 0.02
Molec-biol-promoter	0.45 ± 0.02	0.13 ± 0.08	0.27 ± 0.10	0.28 ± 0.12	0.17 ± 0.13	0.15 ± 0.12	0.15 ± 0.12	0.23 ± 0.09	0.23 ± 0.09	0.30 ± 0.11
Molec-biol-splice	0.40 ± 0.00	0.07 ± 0.01	0.06 ± 0.01	0.25 ± 0.01	0.09 ± 0.01	0.08 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.24 ± 0.02
Monks-1	0.39 ± 0.03	0.37 ± 0.08	0.04 ± 0.07	0.31 ± 0.09	0.19 ± 0.12	0.17 ± 0.11	0.17 ± 0.10	0.14 ± 0.14	0.12 ± 0.16	0.27 ± 0.10
Monks-2	0.47 ± 0.02	0.48 ± 0.03	0.27 ± 0.10	0.39 ± 0.11	0.42 ± 0.07	0.39 ± 0.13	0.36 ± 0.12	0.29 ± 0.09	0.27 ± 0.08	0.41 ± 0.05
Monks-3	0.37 ± 0.03	0.27 ± 0.04	0.11 ± 0.08	0.25 ± 0.15	0.10 ± 0.08	0.11 ± 0.08	0.11 ± 0.08	0.17 ± 0.10	0.13 ± 0.06	0.22 ± 0.04
Mushroom	0.33 ± 0.00	0.12 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00
Musk-1	0.40 ± 0.01	0.26 ± 0.07	0.16 ± 0.07	0.15 ± 0.05	0.09 ± 0.04	0.08 ± 0.03	0.06 ± 0.03	0.07 ± 0.02	0.07 ± 0.02	0.38 ± 0.09
Musk-2	0.20 ± 0.00	0.15 ± 0.02	0.05 ± 0.01	0.04 ± 0.01	0.04 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.10 ± 0.01
Newdiagnosis	0.23 ± 0.02	0.10 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.22 ± 0.07
Nursery	0.24 ± 0.00	0.08 ± 0.00	0.00 ± 0.00	0.05 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.02 ± 0.01	0.03 ± 0.01	0.12 ± 0.00
Obesitydataset-raw-and-data-sinthetic	0.19 ± 0.00	0.11 ± 0.00	0.02 ± 0.01	0.05 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.00	0.09 ± 0.01
Obs-network-dataset-2-aug27	0.15 ± 0.01	0.14 ± 0.03	0.00 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.03 ± 0.01	0.02 ± 0.01	0.05 ± 0.01
Occupancy-data	0.09 ± 0.01	0.05 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.19 ± 0.02	0.18 ± 0.02	0.18 ± 0.02	0.04 ± 0.01	0.03 ± 0.01	0.01 ± 0.01
Occupancy-data2	0.07 ± 0.00	0.04 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.16 ± 0.01	0.14 ± 0.01	0.14 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
Occupancy-data3	0.05 ± 0.00	0.02 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.17 ± 0.01	0.15 ± 0.01	0.15 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
Old	0.42 ± 0.00	0.08 ± 0.01	0.06 ± 0.01	0.03 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.03 ± 0.01	0.03 ± 0.01	0.11 ± 0.02
Online-shoppers-intention	0.24 ± 0.00	0.23 ± 0.01	0.14 ± 0.00	0.18 ± 0.01	0.15 ± 0.00	0.16 ± 0.00	0.16 ± 0.00	0.13 ± 0.01	0.14 ± 0.02	0.12 ± 0.01
Oocytes-merluccius-nucleus-4d	0.39 ± 0.01	0.40 ± 0.04	0.29 ± 0.04	0.28 ± 0.04	0.23 ± 0.03	0.20 ± 0.03	0.19 ± 0.03	0.20 ± 0.03	0.21 ± 0.03	0.33 ± 0.05
Oocytes-merluccius-states-2f	0.17 ± 0.01	0.10 ± 0.02	0.07 ± 0.02	0.06 ± 0.02	0.06 ± 0.02	0.05 ± 0.01	0.05 ± 0.01	0.06 ± 0.01	0.06 ± 0.01	0.12 ± 0.03
Oocytes-trisopterus-nucleus-2f	0.44 ± 0.01	0.46 ± 0.05	0.30 ± 0.05	0.26 ± 0.06	0.19 ± 0.04	0.16 ± 0.03	0.17 ± 0.03	0.18 ± 0.03	0.19 ± 0.03	0.41 ± 0.06
Oocytes-trisopterus-states-5b	0.20 ± 0.02	0.16 ± 0.03	0.09 ± 0.02	0.06 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.13 ± 0.02
Optdigits	0.16 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.15 ± 0.00
Optical	0.16 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.14 ± 0.00
Ozone	0.06 ± 0.00	0.29 ± 0.04	0.05 ± 0.01	0.05 ± 0.01	0.03 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.04 ± 0.01	0.03 ± 0.01	0.03 ± 0.00
Page-blocks	0.05 ± 0.00	0.04 ± 0.01	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
Parkingbirmingham	0.01 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.00 ± 0.00
Parkinsons	0.28 ± 0.03	0.31 ± 0.11	0.20 ± 0.03	0.04 ± 0.02	0.12 ± 0.06	0.10 ± 0.06	0.09 ± 0.06	0.12 ± 0.06	0.11 ± 0.07	0.14 ± 0.08
Pasture	0.37 ± 0.04	0.18 ± 0.15	0.16 ± 0.13	0.21 ± 0.17	0.48 ± 0.03	0.48 ± 0.03	0.48 ± 0.03	0.19 ± 0.16	0.20 ± 0.16	0.24 ± 0.18
Pbc	0.40 ± 0.01	0.24 ± 0.06	0.29 ± 0.06	0.40 ± 0.09	0.39 ± 0.01	0.39 ± 0.01	0.39 ± 0.01	0.29 ± 0.05	0.27 ± 0.03	0.28 ± 0.09
Pen	0.12 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.17 ± 0.00	0.17 ± 0.00	0.17 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.12 ± 0.00
Pendigits	0.12 ± 0.00	0.03 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.12 ± 0.00
Pharynx	0.36 ± 0.02	0.39 ± 0.04	0.36 ± 0.01	0.36 ± 0.10	0.25 ± 0.09	0.27 ± 0.11	0.30 ± 0.11	0.29 ± 0.04	0.30 ± 0.16	0.73 ± 0.06
Phishingwebsites	0.43 ± 0.00	0.09 ± 0.01	0.06 ± 0.00	0.03 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.05 ± 0.00	0.03 ± 0.00	0.03 ± 0.00	0.11 ± 0.01
Pima	0.39 ± 0.01	0.28 ± 0.03	0.32 ± 0.05	0.30 ± 0.05	0.24 ± 0.04	0.25 ± 0.04	0.25 ± 0.03	0.30 ± 0.02	0.29 ± 0.02	0.29 ± 0.05
Pittsburg-bridges-rel-l	0.34 ± 0.02	0.26 ± 0.06	0.31 ± 0.07	0.20 ± 0.11	0.22 ± 0.08	0.21 ± 0.08	0.18 ± 0.07	0.27 ± 0.07	0.26 ± 0.07	0.19 ± 0.07
Pittsburg-bridges-span	0.35 ± 0.01	0.26 ± 0.05	0.30 ± 0.07	0.30 ± 0.10	0.23 ± 0.10	0.24 ± 0.11	0.24 ± 0.11	0.24 ± 0.04	0.22 ± 0.07	0.30 ± 0.09
Pittsburg-bridges-t-or-d	0.22 ± 0.04	0.20 ± 0.07	0.21 ± 0.09	0.17 ± 0.12	0.14 ± 0.08	0.12 ± 0.08	0.11 ± 0.08	0.15 ± 0.12	0.16 ± 0.13	0.15 ± 0.10
Pittsburg-bridges-type	0.23 ± 0.01	0.17 ± 0.02	0.17 ± 0.02	0.15 ± 0.04	0.15 ± 0.04	0.13 ± 0.06	0.13 ± 0.04	0.15 ± 0.05	0.15 ± 0.05	0.15 ± 0.02
Pittsburg-bridgesmaterial	0.21 ± 0.02	0.13 ± 0.05	0.13 ± 0.04	0.12 ± 0.07	0.09 ± 0.04	0.10 ± 0.04	0.11 ± 0.06	0.13 ± 0.06	0.15 ± 0.06	0.09 ± 0.03
Planning	0.41 ± 0.01	0.43 ± 0.03	0.41 ± 0.01	0.34 ± 0.09	0.29 ± 0.02	0.30 ± 0.06	0.34 ± 0.07	0.41 ± 0.05	0.43 ± 0.08	0.38 ± 0.09
Plant-margin	0.02 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00

Table 9: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Plant-shape	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00
Plant-texture	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00
Poker-hand-training-true	0.11 ± 0.00	0.11 ± 0.00	0.10 ± 0.00	0.11 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.09 ± 0.00	0.11 ± 0.00	0.11 ± 0.00	0.10 ± 0.00
Post-operative	0.28 ± 0.02	0.30 ± 0.04	0.28 ± 0.02	0.29 ± 0.07	0.19 ± 0.04	0.21 ± 0.05	0.22 ± 0.06	0.28 ± 0.07	0.29 ± 0.08	0.21 ± 0.05
Primary-tumor	0.11 ± 0.00	0.08 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.07 ± 0.02	0.08 ± 0.01	0.08 ± 0.01	0.10 ± 0.00
Qsarbioconcentration	0.38 ± 0.00	0.38 ± 0.02	0.39 ± 0.02	0.38 ± 0.06	0.25 ± 0.01	0.27 ± 0.02	0.29 ± 0.03	0.39 ± 0.02	0.39 ± 0.02	0.27 ± 0.03
Qsarbiodegradation	0.39 ± 0.01	0.24 ± 0.03	0.19 ± 0.03	0.16 ± 0.02	0.14 ± 0.03	0.14 ± 0.02	0.14 ± 0.02	0.14 ± 0.02	0.14 ± 0.02	0.23 ± 0.04
Qualitative-bankruptcy	0.13 ± 0.01	0.01 ± 0.02	0.02 ± 0.03	0.00 ± 0.00	0.01 ± 0.03	0.01 ± 0.03	0.00 ± 0.01	0.01 ± 0.02	0.01 ± 0.02	0.02 ± 0.03
Ringnorm	0.30 ± 0.01	0.02 ± 0.00	0.09 ± 0.00	0.25 ± 0.02	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.08 ± 0.01	0.09 ± 0.01	0.36 ± 0.01
Risk-factors-cervical-cancer	0.12 ± 0.01	0.11 ± 0.01	0.06 ± 0.02	0.06 ± 0.02	0.06 ± 0.01	0.07 ± 0.01	0.06 ± 0.02	0.05 ± 0.02	0.05 ± 0.02	0.04 ± 0.02
Robotnavigation	0.16 ± 0.00	0.24 ± 0.01	0.00 ± 0.00	0.06 ± 0.01	0.05 ± 0.01	0.04 ± 0.00	0.04 ± 0.00	0.07 ± 0.01	0.06 ± 0.01	0.12 ± 0.01
Sapfile	0.41 ± 0.01	0.33 ± 0.04	0.40 ± 0.08	0.41 ± 0.06	0.34 ± 0.06	0.34 ± 0.05	0.34 ± 0.06	0.35 ± 0.05	0.35 ± 0.07	0.39 ± 0.08
Sat	0.14 ± 0.00	0.07 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	0.25 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.13 ± 0.00
Satelite	0.14 ± 0.00	0.07 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	0.25 ± 0.00	0.24 ± 0.00	0.24 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.13 ± 0.01
Scadi	0.18 ± 0.01	0.05 ± 0.02	0.07 ± 0.03	0.08 ± 0.04	0.07 ± 0.03	0.06 ± 0.04	0.05 ± 0.04	0.06 ± 0.03	0.06 ± 0.03	0.12 ± 0.03
Schillingdata	0.19 ± 0.00	0.10 ± 0.01	0.23 ± 0.00	0.15 ± 0.01	0.13 ± 0.00	0.11 ± 0.01	0.08 ± 0.01	0.07 ± 0.02	0.09 ± 0.03	0.13 ± 0.00
Seeds	0.19 ± 0.03	0.06 ± 0.03	0.07 ± 0.04	0.04 ± 0.01	0.04 ± 0.03	0.04 ± 0.03	0.04 ± 0.03	0.04 ± 0.01	0.04 ± 0.01	0.12 ± 0.07
Segment	0.13 ± 0.00	0.06 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.10 ± 0.01	0.09 ± 0.01	0.09 ± 0.01	0.02 ± 0.00	0.01 ± 0.00	0.10 ± 0.01
Seismic-bumps	0.12 ± 0.00	0.14 ± 0.02	0.12 ± 0.00	0.11 ± 0.01	0.07 ± 0.00	0.07 ± 0.00	0.07 ± 0.00	0.10 ± 0.01	0.09 ± 0.01	0.07 ± 0.01
Semeion	0.17 ± 0.00	0.03 ± 0.01	0.05 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.16 ± 0.00
Setapprocesst1	0.40 ± 0.03	0.42 ± 0.19	0.38 ± 0.18	0.35 ± 0.10	0.36 ± 0.08	0.31 ± 0.11	0.33 ± 0.09	0.36 ± 0.19	0.37 ± 0.19	0.28 ± 0.23
Setapprocesst10	0.44 ± 0.02	0.56 ± 0.15	0.42 ± 0.14	0.40 ± 0.16	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.47 ± 0.24	0.47 ± 0.21	0.38 ± 0.08
Setapprocesst11	0.44 ± 0.02	0.45 ± 0.25	0.36 ± 0.17	0.42 ± 0.16	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.39 ± 0.15	0.35 ± 0.16	0.43 ± 0.11
Setapprocesst2	0.44 ± 0.02	0.33 ± 0.15	0.27 ± 0.18	0.38 ± 0.21	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.38 ± 0.17	0.35 ± 0.17	0.19 ± 0.17
Setapprocesst3	0.44 ± 0.01	0.35 ± 0.12	0.40 ± 0.15	0.34 ± 0.13	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.35 ± 0.12	0.35 ± 0.08	0.47 ± 0.16
Setapprocesst4	0.41 ± 0.03	0.51 ± 0.24	0.45 ± 0.14	0.40 ± 0.18	0.29 ± 0.07	0.29 ± 0.07	0.29 ± 0.07	0.45 ± 0.19	0.43 ± 0.16	0.27 ± 0.12
Setapprocesst5	0.43 ± 0.03	0.53 ± 0.17	0.47 ± 0.17	0.41 ± 0.17	0.35 ± 0.08	0.36 ± 0.09	0.36 ± 0.09	0.41 ± 0.13	0.39 ± 0.13	0.34 ± 0.25
Setapprocesst6	0.44 ± 0.01	0.33 ± 0.16	0.35 ± 0.12	0.36 ± 0.14	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.37 ± 0.15	0.37 ± 0.14	0.40 ± 0.20
Setapprocesst7	0.44 ± 0.03	0.26 ± 0.14	0.39 ± 0.20	0.36 ± 0.15	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.28 ± 0.18	0.29 ± 0.18	0.47 ± 0.15
Setapprocesst8	0.43 ± 0.02	0.32 ± 0.14	0.30 ± 0.15	0.37 ± 0.15	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.16	0.34 ± 0.17	0.51 ± 0.11
Setapprocesst9	0.44 ± 0.02	0.35 ± 0.15	0.32 ± 0.16	0.42 ± 0.10	0.34 ± 0.06	0.34 ± 0.06	0.34 ± 0.06	0.29 ± 0.18	0.33 ± 0.16	0.49 ± 0.17
Shillbiddingdataset	0.12 ± 0.00	0.04 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.03 ± 0.01
Shuttle-landing-control	0.16 ± 0.17	0.23 ± 0.29	0.12 ± 0.13	0.14 ± 0.19	0.05 ± 0.16	0.05 ± 0.16	0.05 ± 0.16	0.16 ± 0.27	0.16 ± 0.29	0.15 ± 0.34
Somervillehappinesssurvey2015	0.47 ± 0.03	0.44 ± 0.06	0.41 ± 0.10	0.42 ± 0.10	0.40 ± 0.11	0.41 ± 0.10	0.41 ± 0.12	0.42 ± 0.11	0.43 ± 0.11	0.34 ± 0.16
Sonar	0.45 ± 0.01	0.31 ± 0.09	0.29 ± 0.07	0.14 ± 0.07	0.34 ± 0.05	0.30 ± 0.09	0.26 ± 0.10	0.19 ± 0.10	0.19 ± 0.09	0.37 ± 0.11
Soybean	0.09 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.01 ± 0.00	0.01 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.08 ± 0.00
Spambase	0.40 ± 0.00	0.21 ± 0.02	0.09 ± 0.01	0.09 ± 0.01	0.07 ± 0.01	0.06 ± 0.01	0.07 ± 0.01	0.11 ± 0.01	0.13 ± 0.06	0.22 ± 0.02
Speaker-accent	0.21 ± 0.01	0.15 ± 0.03	0.11 ± 0.03	0.07 ± 0.03	0.14 ± 0.01	0.13 ± 0.01	0.13 ± 0.01	0.08 ± 0.02	0.09 ± 0.02	0.17 ± 0.02
Spect	0.42 ± 0.03	0.31 ± 0.14	0.38 ± 0.09	0.49 ± 0.12	0.35 ± 0.11	0.36 ± 0.12	0.37 ± 0.09	0.46 ± 0.17	0.47 ± 0.16	0.28 ± 0.10
Spectf	0.47 ± 0.02	0.25 ± 0.12	0.28 ± 0.11	0.33 ± 0.18	0.32 ± 0.06	0.32 ± 0.06	0.24 ± 0.14	0.30 ± 0.14	0.29 ± 0.13	0.29 ± 0.08
Statlog-australian-credit	0.43 ± 0.00	0.46 ± 0.02	0.41 ± 0.05	0.44 ± 0.03	0.32 ± 0.02	0.36 ± 0.06	0.36 ± 0.06	0.41 ± 0.03	0.41 ± 0.03	0.35 ± 0.06
Statlog-german-credit	0.41 ± 0.00	0.29 ± 0.03	0.31 ± 0.04	0.32 ± 0.04	0.24 ± 0.04	0.23 ± 0.04	0.23 ± 0.04	0.30 ± 0.04	0.28 ± 0.03	0.29 ± 0.02
Statlog-heart	0.40 ± 0.02	0.19 ± 0.06	0.28 ± 0.08	0.25 ± 0.08	0.17 ± 0.05	0.20 ± 0.08	0.20 ± 0.10	0.22 ± 0.06	0.21 ± 0.08	0.29 ± 0.06
Statlog-image	0.12 ± 0.00	0.06 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.10 ± 0.01

Table 9: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Statlog-landsat	0.14 ± 0.00	0.07 ± 0.01	0.05 ± 0.01	0.03 ± 0.01	0.04 ± 0.01	0.03 ± 0.00	0.03 ± 0.01	0.04 ± 0.00	0.04 ± 0.01	0.13 ± 0.01
Statlog-shuttle	0.07 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.00
Statlog-vehicle	0.29 ± 0.01	0.28 ± 0.01	0.14 ± 0.02	0.15 ± 0.02	0.12 ± 0.02	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.02	0.24 ± 0.03
Steel-plates	0.17 ± 0.00	0.12 ± 0.01	0.07 ± 0.01	0.08 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.15 ± 0.01
Synthetic-control	0.19 ± 0.00	0.02 ± 0.01	0.03 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.14 ± 0.01
Teaching	0.40 ± 0.02	0.38 ± 0.03	0.29 ± 0.07	0.25 ± 0.09	0.29 ± 0.10	0.26 ± 0.08	0.25 ± 0.08	0.36 ± 0.03	0.36 ± 0.03	0.36 ± 0.07
Thoraricsurgery	0.25 ± 0.00	0.28 ± 0.08	0.25 ± 0.00	0.23 ± 0.06	0.15 ± 0.00	0.16 ± 0.01	0.16 ± 0.01	0.24 ± 0.04	0.22 ± 0.04	0.17 ± 0.02
Thyroid	0.18 ± 0.03	0.02 ± 0.02	0.06 ± 0.04	0.03 ± 0.02	0.16 ± 0.03	0.15 ± 0.03	0.15 ± 0.03	0.03 ± 0.02	0.03 ± 0.02	0.06 ± 0.06
Thyroid-train	0.09 ± 0.00	0.04 ± 0.01	0.00 ± 0.00	0.05 ± 0.01	0.03 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.03 ± 0.01	0.03 ± 0.01	0.02 ± 0.01
Tic-tac-toe	0.44 ± 0.00	0.37 ± 0.01	0.07 ± 0.02	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.30 ± 0.02
Titanic	0.37 ± 0.01	0.31 ± 0.02	0.31 ± 0.01	0.31 ± 0.01	0.22 ± 0.02	0.21 ± 0.02	0.21 ± 0.02	0.32 ± 0.02	0.32 ± 0.03	0.22 ± 0.02
Trains	0.46 ± 0.08	0.40 ± 0.52	0.28 ± 0.46	0.42 ± 0.42	0.40 ± 0.52	0.30 ± 0.48	0.30 ± 0.48	0.28 ± 0.37	0.27 ± 0.37	1.00 ± 0.00
Transfusion	0.35 ± 0.01	0.29 ± 0.02	0.30 ± 0.01	0.32 ± 0.04	0.25 ± 0.03	0.27 ± 0.03	0.28 ± 0.04	0.30 ± 0.02	0.30 ± 0.02	0.24 ± 0.02
Trial	0.32 ± 0.02	0.10 ± 0.03	0.00 ± 0.00	0.00 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00
Turkiye-student-evaluation	0.34 ± 0.00	0.10 ± 0.01	0.00 ± 0.00	0.06 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Unbalanced	0.03 ± 0.00	0.11 ± 0.04	0.03 ± 0.00	0.02 ± 0.02	0.01 ± 0.00	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.02 ± 0.01	0.01 ± 0.00
Urbanlandcover	0.16 ± 0.01	0.05 ± 0.02	0.05 ± 0.01	0.06 ± 0.02	0.18 ± 0.00	0.18 ± 0.00	0.18 ± 0.00	0.05 ± 0.02	0.05 ± 0.02	0.11 ± 0.03
Userknowledgemodeling	0.27 ± 0.02	0.11 ± 0.02	0.04 ± 0.02	0.10 ± 0.05	0.11 ± 0.03	0.07 ± 0.02	0.07 ± 0.02	0.05 ± 0.02	0.04 ± 0.02	0.07 ± 0.04
Vehicle	0.29 ± 0.01	0.28 ± 0.01	0.14 ± 0.02	0.15 ± 0.02	0.35 ± 0.01	0.34 ± 0.02	0.34 ± 0.02	0.10 ± 0.01	0.10 ± 0.02	0.24 ± 0.03
Vertebral-column-2classes	0.34 ± 0.04	0.22 ± 0.07	0.21 ± 0.06	0.18 ± 0.05	0.15 ± 0.05	0.15 ± 0.06	0.15 ± 0.06	0.18 ± 0.04	0.18 ± 0.04	0.26 ± 0.08
Vertebral-column-3classes	0.28 ± 0.02	0.14 ± 0.02	0.13 ± 0.04	0.15 ± 0.03	0.11 ± 0.04	0.10 ± 0.05	0.11 ± 0.04	0.13 ± 0.02	0.13 ± 0.02	0.17 ± 0.05
Veteran	0.39 ± 0.01	0.31 ± 0.05	0.34 ± 0.04	0.40 ± 0.09	0.29 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.39 ± 0.12	0.34 ± 0.12	0.26 ± 0.08
Vowel	0.13 ± 0.00	0.09 ± 0.01	0.04 ± 0.01	0.00 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.02 ± 0.01	0.02 ± 0.00	0.12 ± 0.01
Wall-following	0.16 ± 0.00	0.24 ± 0.00	0.00 ± 0.00	0.06 ± 0.01	0.05 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.12 ± 0.01
Waveform-noise	0.37 ± 0.00	0.14 ± 0.01	0.17 ± 0.01	0.18 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.31 ± 0.02
Waveform	0.34 ± 0.00	0.13 ± 0.01	0.17 ± 0.01	0.15 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.11 ± 0.01	0.11 ± 0.01	0.32 ± 0.01
Wbc	0.10 ± 0.01	0.03 ± 0.01	0.09 ± 0.02	0.05 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.04 ± 0.01	0.04 ± 0.02	0.09 ± 0.03
Wdbc	0.25 ± 0.02	0.07 ± 0.04	0.07 ± 0.04	0.04 ± 0.02	0.37 ± 0.01	0.37 ± 0.01	0.37 ± 0.01	0.03 ± 0.02	0.04 ± 0.02	0.10 ± 0.03
Weathernominal	0.45 ± 0.15	0.42 ± 0.16	0.36 ± 0.34	0.48 ± 0.24	0.30 ± 0.35	0.40 ± 0.39	0.25 ± 0.35	0.25 ± 0.35	0.25 ± 0.35	0.55 ± 0.44
Weathernumeric	0.55 ± 0.16	0.45 ± 0.18	0.24 ± 0.33	0.24 ± 0.30	0.30 ± 0.35	0.50 ± 0.47	0.50 ± 0.47	0.28 ± 0.39	0.29 ± 0.40	0.55 ± 0.44
Website-phishingdata	0.29 ± 0.00	0.14 ± 0.01	0.08 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.09 ± 0.01	0.08 ± 0.01	0.09 ± 0.01	0.08 ± 0.01	0.12 ± 0.01
Wholesalecustomersdata	0.23 ± 0.02	0.12 ± 0.03	0.13 ± 0.03	0.12 ± 0.05	0.32 ± 0.01	0.32 ± 0.01	0.32 ± 0.01	0.11 ± 0.03	0.11 ± 0.03	0.10 ± 0.04
Wifi-localization	0.20 ± 0.00	0.01 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.01	0.01 ± 0.00	0.01 ± 0.00	0.11 ± 0.02
Wilt	0.10 ± 0.00	0.17 ± 0.01	0.02 ± 0.00	0.05 ± 0.01	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.03 ± 0.00	0.02 ± 0.00	0.06 ± 0.00
Wine-quality-red	0.18 ± 0.00	0.18 ± 0.01	0.14 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.17 ± 0.01	0.16 ± 0.01	0.15 ± 0.01
Wine-quality-white	0.17 ± 0.00	0.17 ± 0.00	0.13 ± 0.01	0.10 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.12 ± 0.01	0.16 ± 0.00	0.16 ± 0.00	0.15 ± 0.01
Wine	0.23 ± 0.01	0.02 ± 0.03	0.05 ± 0.03	0.04 ± 0.03	0.01 ± 0.02	0.01 ± 0.02	0.01 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.15 ± 0.06
Yamilnaduelectricty	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.08 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00	0.09 ± 0.00
Yeast	0.14 ± 0.00	0.10 ± 0.00	0.10 ± 0.01	0.10 ± 0.00	0.08 ± 0.01	0.08 ± 0.01	0.08 ± 0.01	0.10 ± 0.00	0.10 ± 0.00	0.12 ± 0.01
Youtobe-kabita-preprocessing	0.23 ± 0.00	0.21 ± 0.00	0.21 ± 0.00	0.20 ± 0.00	0.18 ± 0.01	0.18 ± 0.01	0.18 ± 0.01	0.22 ± 0.00	0.21 ± 0.00	0.19 ± 0.01
Youtobe-nisha-preprocessing	0.22 ± 0.00	0.21 ± 0.00	0.20 ± 0.00	0.20 ± 0.00	0.17 ± 0.01	0.17 ± 0.01	0.17 ± 0.01	0.21 ± 0.00	0.21 ± 0.00	0.20 ± 0.00
Z-alizadehsani	0.39 ± 0.01	0.20 ± 0.06	0.24 ± 0.08	0.22 ± 0.07	0.29 ± 0.01	0.29 ± 0.01	0.29 ± 0.01	0.17 ± 0.07	0.18 ± 0.06	0.29 ± 0.05
Zoo	0.14 ± 0.01	0.01 ± 0.02	0.02 ± 0.02	0.02 ± 0.01	0.02 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.02 ± 0.02	0.08 ± 0.03
Average mae (rank)	0.2672 (10)	0.1947 (9)	0.1550 (3)	0.1576 (4)	0.1695 (7)	0.1662 (6)	0.1643 (5)	0.1494 (2)	0.1476 (1)	0.1913 (8)
Average mae std (rank)	0.0120 (1)	0.0374 (6)	0.0372 (5)	0.0431 (9)	0.0318 (2)	0.0345 (3)	0.0350 (4)	0.0400 (7)	0.0413 (8)	0.0438 (10)

Table 10: The detailed experimental results on relative absolute error and standard deviation.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Abalone	75.92 ± 1.42	64.51 ± 3.17	64.97 ± 1.91	63.17 ± 3.75	50.10 ± 3.70	49.56 ± 3.54	49.63 ± 3.38	65.50 ± 2.15	64.96 ± 2.45	58.61 ± 2.68
Absenteeism-at-work	62.76 ± 1.26	33.47 ± 5.17	0.73 ± 1.25	33.50 ± 5.85	31.11 ± 6.09	31.11 ± 6.09	31.11 ± 6.09	7.25 ± 2.91	7.28 ± 3.18	9.63 ± 3.49
Acute-inflammation	46.10 ± 4.81	19.62 ± 5.03	0.00 ± 0.00	1.64 ± 0.10	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.10 ± 0.10	0.79 ± 0.08	41.67 ± 16.20
Acute-nephritis	43.11 ± 2.40	10.82 ± 11.20	0.00 ± 0.00	1.66 ± 0.10	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.12 ± 0.14	0.90 ± 0.18	17.13 ± 16.15
Adult	94.23 ± 1.18	55.36 ± 6.78	60.24 ± 7.54	59.42 ± 9.83	45.84 ± 11.29	45.50 ± 11.36	45.67 ± 11.76	58.87 ± 7.66	58.37 ± 7.45	53.92 ± 3.56
Aggregation	17.56 ± 6.47	1.06 ± 0.54	0.48 ± 0.78	1.23 ± 0.52	0.16 ± 0.52	0.33 ± 0.69	0.33 ± 0.69	7.37 ± 2.22	6.05 ± 2.49	51.54 ± 3.72
Algerianforest	61.49 ± 4.72	12.11 ± 10.35	9.21 ± 8.93	24.91 ± 13.53	83.64 ± 3.36	77.76 ± 8.52	77.76 ± 8.52	8.33 ± 6.21	7.25 ± 7.42	5.78 ± 9.62
Annealing	91.67 ± 0.60	76.69 ± 9.34	10.96 ± 5.44	27.17 ± 5.51	27.39 ± 7.85	21.16 ± 6.34	19.61 ± 5.29	25.69 ± 8.98	24.05 ± 6.96	40.47 ± 0.10
Arrhythmia	96.40 ± 0.43	56.91 ± 10.57	56.76 ± 10.48	70.08 ± 4.75	56.02 ± 3.65	50.52 ± 5.06	51.82 ± 6.11	50.80 ± 7.65	66.27 ± 20.92	59.26 ± 3.73
Au1-1000	99.01 ± 0.36	92.85 ± 2.65	77.84 ± 7.92	84.06 ± 11.34	67.43 ± 0.59	67.43 ± 0.59	66.65 ± 2.15	81.90 ± 9.76	75.90 ± 9.28	67.43 ± 0.59
Au4-2500	95.13 ± 0.73	83.45 ± 2.61	52.13 ± 9.35	64.38 ± 7.72	67.92 ± 4.82	67.92 ± 4.82	67.92 ± 4.82	62.96 ± 6.80	62.81 ± 7.92	81.84 ± 7.39
Au6-1000	99.65 ± 0.37	99.20 ± 0.74	94.38 ± 4.26	100.82 ± 4.36	88.47 ± 0.01	88.47 ± 0.01	88.47 ± 0.01	99.67 ± 3.16	100.64 ± 4.46	88.13 ± 2.58
Au6-250-drift-au6-cd1-500	99.81 ± 0.54	98.95 ± 1.65	95.38 ± 4.14	97.98 ± 4.88	91.22 ± 0.76	91.22 ± 0.76	91.22 ± 0.76	99.52 ± 2.11	98.83 ± 3.99	88.26 ± 4.02
Au6-cd1-400	99.51 ± 0.58	95.72 ± 3.69	77.36 ± 8.00	99.20 ± 9.43	87.01 ± 0.82	87.01 ± 0.82	87.01 ± 0.82	97.11 ± 6.08	97.61 ± 5.16	82.20 ± 7.92
Au7-300-drift-au7-cpd1-800	98.16 ± 0.61	92.66 ± 1.38	81.78 ± 4.66	86.32 ± 5.39	90.85 ± 2.08	91.77 ± 2.45	91.77 ± 2.45	88.86 ± 2.66	87.74 ± 3.17	95.23 ± 3.94
Au7-700	97.73 ± 1.32	91.56 ± 2.35	75.84 ± 5.04	96.84 ± 8.31	94.08 ± 4.32	95.37 ± 3.52	95.37 ± 3.52	90.41 ± 3.23	88.33 ± 3.98	82.94 ± 5.72
Au7-cpd1-500	98.68 ± 0.92	94.29 ± 2.51	70.29 ± 5.31	83.29 ± 7.52	82.49 ± 1.33	81.96 ± 2.13	81.96 ± 2.13	84.23 ± 5.60	82.95 ± 6.86	79.82 ± 6.44
Audiology-std	96.20 ± 0.35	31.94 ± 11.68	29.80 ± 8.89	38.14 ± 12.51	42.68 ± 9.22	37.28 ± 11.79	33.87 ± 9.68	25.65 ± 7.03	26.42 ± 5.98	60.90 ± 2.39
Audit-risk	58.08 ± 2.67	11.29 ± 4.10	0.27 ± 0.85	4.85 ± 2.54	3.24 ± 2.78	2.69 ± 2.83	2.42 ± 2.95	8.64 ± 4.01	8.72 ± 3.90	0.00 ± 0.00
Autism-adolescent-data	77.89 ± 5.47	9.38 ± 6.01	0.00 ± 0.00	21.64 ± 13.14	8.02 ± 10.37	9.94 ± 10.50	8.02 ± 10.37	21.09 ± 18.16	20.85 ± 18.51	0.00 ± 0.00
Autism-adult-data	74.39 ± 2.64	8.31 ± 4.77	0.00 ± 0.00	13.34 ± 9.74	1.81 ± 2.55	1.09 ± 2.45	1.45 ± 2.53	0.56 ± 0.31	0.44 ± 0.25	0.00 ± 0.00
Autism-child-data	79.33 ± 2.43	8.93 ± 3.35	0.00 ± 0.00	23.54 ± 7.65	0.69 ± 2.18	0.00 ± 0.00	0.00 ± 0.00	1.82 ± 1.73	1.95 ± 1.88	0.00 ± 0.00
Autos	75.47 ± 3.50	56.55 ± 14.32	26.24 ± 8.52	33.42 ± 12.55	84.46 ± 5.52	83.20 ± 6.66	83.20 ± 6.66	31.92 ± 7.36	31.38 ± 6.33	48.67 ± 12.86
Avila	51.66 ± 0.55	97.65 ± 2.37	8.75 ± 1.60	31.65 ± 3.48	41.49 ± 2.74	38.68 ± 2.40	36.83 ± 2.97	62.83 ± 2.84	62.56 ± 3.53	36.74 ± 3.26
Balance-scale	73.87 ± 2.73	56.05 ± 2.27	49.81 ± 6.48	33.52 ± 7.43	16.85 ± 2.67	12.64 ± 1.52	11.24 ± 2.34	21.20 ± 2.88	21.89 ± 3.60	76.65 ± 7.70
Balloons	81.25 ± 23.98	100.00 ± 0.00	91.06 ± 31.00	68.48 ± 22.88	57.00 ± 63.60	37.00 ± 62.55	37.00 ± 62.55	56.90 ± 46.90	54.78 ± 48.74	112.50 ± 77.28
Bank	95.82 ± 0.54	87.54 ± 4.70	70.70 ± 4.16	68.26 ± 6.54	51.27 ± 3.03	50.51 ± 2.78	49.86 ± 5.20	63.84 ± 5.63	65.32 ± 9.20	56.04 ± 3.01
Blood	96.32 ± 1.56	79.63 ± 5.50	83.64 ± 3.41	88.71 ± 11.09	59.27 ± 5.65	57.81 ± 4.24	59.28 ± 3.90	81.45 ± 4.21	81.29 ± 5.46	65.18 ± 4.59
Breast-cancer-wisc-diag	52.63 ± 4.65	14.91 ± 8.87	15.46 ± 8.53	9.03 ± 3.99	5.25 ± 4.74	4.51 ± 4.26	3.76 ± 2.50	8.15 ± 3.78	8.12 ± 4.00	23.28 ± 8.76
Breast-cancer-wisc-prog	97.66 ± 1.44	93.25 ± 26.65	79.33 ± 16.09	75.56 ± 24.57	56.86 ± 16.40	60.03 ± 25.23	61.33 ± 27.19	75.38 ± 25.65	73.70 ± 30.26	82.38 ± 22.18
Breast-cancer-wisc	30.02 ± 4.69	9.08 ± 3.74	16.30 ± 7.31	10.02 ± 6.58	6.33 ± 4.22	6.65 ± 4.34	7.28 ± 4.22	12.22 ± 7.57	10.45 ± 4.70	16.14 ± 10.05
Breast-cancer	93.44 ± 2.87	75.65 ± 9.99	87.28 ± 6.25	72.06 ± 18.00	62.49 ± 12.22	64.17 ± 12.75	66.71 ± 12.25	77.44 ± 13.67	78.80 ± 8.96	70.83 ± 11.74
Breast-tissue	67.58 ± 7.19	40.04 ± 13.68	41.08 ± 16.69	38.07 ± 15.80	50.18 ± 17.49	38.79 ± 19.93	36.37 ± 20.11	51.89 ± 8.28	51.91 ± 11.17	54.78 ± 11.74
Bupa	92.74 ± 3.01	94.88 ± 6.47	81.13 ± 16.90	78.64 ± 10.05	84.41 ± 4.54	83.81 ± 4.89	83.81 ± 4.89	76.65 ± 7.63	73.89 ± 7.61	93.87 ± 16.05
Caesarian	88.07 ± 11.68	84.08 ± 21.17	93.33 ± 11.98	96.10 ± 29.64	91.75 ± 12.23	76.84 ± 25.03	84.51 ± 35.13	91.04 ± 37.86	91.03 ± 36.15	105.03 ± 38.01
Car	94.52 ± 0.77	54.16 ± 2.40	6.32 ± 1.72	8.13 ± 2.30	5.69 ± 2.47	4.17 ± 2.52	3.66 ± 2.10	18.40 ± 2.78	17.59 ± 3.27	65.45 ± 0.26
Cardiotocography-10classes	81.53 ± 0.59	36.58 ± 2.75	23.93 ± 2.57	25.86 ± 3.51	22.35 ± 2.49	20.95 ± 3.05	20.11 ± 2.46	24.50 ± 3.11	25.72 ± 2.69	62.40 ± 2.72
Cardiotocography-3classes	87.86 ± 1.25	50.02 ± 8.43	23.36 ± 6.87	21.95 ± 4.73	24.26 ± 5.22	21.57 ± 5.64	21.19 ± 4.73	23.56 ± 2.74	22.59 ± 5.56	50.44 ± 8.62
Cervical-cancer	82.50 ± 7.36	23.68 ± 18.11	38.98 ± 28.40	30.08 ± 26.38	69.87 ± 5.37	62.97 ± 15.91	62.97 ± 15.91	18.32 ± 18.30	18.11 ± 18.48	52.95 ± 20.63
Chemicalcomposionofceramic	48.22 ± 4.83	0.00 ± 0.00	2.50 ± 7.91	2.46 ± 0.01	108.77 ± 4.62	108.77 ± 4.62	108.77 ± 4.62	1.03 ± 0.45	0.76 ± 0.36	0.00 ± 0.00
Chess-krvk	96.17 ± 0.23	90.50 ± 0.64	59.18 ± 2.87	56.30 ± 3.59	60.36 ± 2.90	57.89 ± 2.50	56.18 ± 3.27	74.77 ± 1.16	75.03 ± 1.79	83.17 ± 2.35
Chess-krvkp	94.95 ± 0.25	42.83 ± 3.05	2.14 ± 0.93	20.43 ± 2.78	2.63 ± 1.81	1.63 ± 1.26	1.63 ± 1.07	1.99 ± 1.11	1.82 ± 1.11	67.22 ± 2.90
Congressional-voting	98.94 ± 0.62	93.83 ± 8.29	96.58 ± 5.20	94.06 ± 9.83	80.94 ± 8.39	81.91 ± 11.00	82.87 ± 12.24	91.50 ± 5.95	91.17 ± 6.95	78.06 ± 5.53
Conn-bench-sonar-mines-rocks	90.85 ± 2.57	62.64 ± 17.90	57.49 ± 14.47	27.74 ± 13.91	31.80 ± 14.15	26.05 ± 15.63	23.14 ± 15.75	39.52 ± 18.34	36.71 ± 17.20	73.43 ± 21.12
Conn-bench-vowel-deterding	69.18 ± 0.74	47.05 ± 4.24	24.61 ± 6.91	3.07 ± 1.42	6.46 ± 2.08	2.71 ± 1.97	1.66 ± 1.64	22.53 ± 3.06	23.12 ± 3.66	71.04 ± 5.43
Connect-4	99.54 ± 0.11	100.96 ± 2.40	64.33 ± 4.35	75.45 ± 7.14	67.46 ± 0.66	63.17 ± 2.51	60.13 ± 4.59	70.08 ± 5.04	67.21 ± 4.00	67.23 ± 0.35

Table 10: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Connectionist	90.55 ± 2.58	64.31 ± 21.22	53.66 ± 14.58	26.82 ± 11.63	77.57 ± 17.04	60.81 ± 23.09	54.92 ± 24.46	34.74 ± 15.76	33.84 ± 15.80	74.66 ± 13.93
Contrac	96.44 ± 0.59	85.85 ± 3.75	79.77 ± 4.22	88.33 ± 5.68	70.39 ± 3.42	70.59 ± 5.45	71.54 ± 5.97	80.87 ± 3.06	80.04 ± 2.48	80.47 ± 3.12
Covid-19	82.03 ± 14.93	74.56 ± 43.25	59.63 ± 65.09	79.44 ± 49.06	62.70 ± 59.12	73.88 ± 73.34	59.34 ± 71.97	58.59 ± 61.89	56.92 ± 63.13	30.51 ± 52.42
Credit-approval	83.51 ± 2.47	45.92 ± 6.36	38.63 ± 6.36	36.30 ± 11.38	29.63 ± 11.00	30.21 ± 9.77	31.10 ± 9.29	31.95 ± 5.23	34.02 ± 7.16	29.34 ± 9.07
Crowdsource	73.96 ± 0.93	39.44 ± 4.39	26.14 ± 3.03	10.15 ± 1.97	50.77 ± 2.17	50.77 ± 2.17	50.77 ± 2.17	17.46 ± 3.77	17.68 ± 4.13	44.23 ± 4.55
Crx	82.23 ± 2.36	45.10 ± 5.62	38.94 ± 6.45	38.35 ± 9.75	89.49 ± 7.22	88.02 ± 8.61	85.38 ± 10.37	36.33 ± 8.22	33.45 ± 4.95	29.34 ± 9.07
Cryother	63.26 ± 9.90	36.78 ± 21.94	21.18 ± 14.92	22.05 ± 16.12	35.58 ± 25.91	33.34 ± 26.00	33.34 ± 26.00	27.40 ± 16.86	33.55 ± 18.89	37.93 ± 21.23
Cylinder-bands	93.48 ± 1.64	68.93 ± 11.88	60.97 ± 12.45	64.94 ± 15.13	50.87 ± 8.08	43.91 ± 9.21	39.39 ± 6.58	57.23 ± 10.13	55.07 ± 9.63	70.12 ± 9.03
Dbworld-bodies	97.83 ± 0.70	49.39 ± 31.18	42.52 ± 32.62	82.56 ± 20.22	91.52 ± 10.51	91.52 ± 10.51	72.85 ± 25.44	/	/	27.35 ± 26.75
Dbworld-bodies-stemmed	97.66 ± 0.79	47.11 ± 26.12	30.81 ± 26.46	73.31 ± 28.56	91.52 ± 10.51	82.41 ± 18.05	52.77 ± 26.76	/	/	51.81 ± 41.11
Dbworld-subjects	98.00 ± 0.72	34.71 ± 20.89	61.24 ± 20.24	50.30 ± 24.75	91.52 ± 10.51	91.52 ± 10.51	91.52 ± 10.51	26.89 ± 24.71	26.72 ± 27.31	75.22 ± 24.75
Dbworld-subjects-stemmed	98.01 ± 0.91	34.39 ± 22.86	54.43 ± 27.54	38.63 ± 18.38	91.52 ± 10.51	91.52 ± 10.51	91.52 ± 10.51	28.03 ± 23.32	28.47 ± 23.09	78.63 ± 20.43
Dermatology	72.96 ± 0.82	3.58 ± 3.29	6.40 ± 3.79	7.57 ± 5.23	3.06 ± 2.96	2.71 ± 3.10	2.71 ± 3.49	4.35 ± 1.62	4.10 ± 1.71	62.88 ± 2.99
Diabetes	86.31 ± 3.39	62.50 ± 7.63	69.45 ± 9.95	65.77 ± 10.36	76.77 ± 0.63	76.77 ± 0.63	76.77 ± 0.63	64.99 ± 5.29	63.83 ± 4.80	62.74 ± 11.34
Diabetic	92.32 ± 1.49	86.85 ± 4.44	76.21 ± 7.00	77.67 ± 9.51	83.89 ± 9.59	81.45 ± 12.08	81.27 ± 11.48	66.22 ± 6.30	65.45 ± 5.31	93.84 ± 11.41
Divorce	19.50 ± 5.15	4.71 ± 6.08	11.74 ± 11.75	5.11 ± 5.53	4.71 ± 6.08	4.71 ± 6.08	4.71 ± 6.08	4.07 ± 5.11	4.05 ± 5.17	9.41 ± 10.80
Dota2train	99.84 ± 0.08	94.52 ± 3.62	94.55 ± 8.29	89.94 ± 6.06	94.88 ± 0.69	92.95 ± 7.23	95.10 ± 6.60	99.93 ± 6.20	96.66 ± 4.21	89.49 ± 7.29
Dow-jones-index	89.78 ± 3.12	99.48 ± 6.00	57.59 ± 7.94	90.96 ± 12.70	96.02 ± 0.43	96.02 ± 0.43	96.02 ± 0.43	91.60 ± 6.27	89.30 ± 5.10	87.20 ± 6.17
Dry-bean-dataset	26.66 ± 0.80	12.10 ± 1.59	9.98 ± 1.33	6.74 ± 1.36	46.84 ± 3.31	46.84 ± 3.31	46.84 ± 3.31	11.68 ± 1.87	11.20 ± 1.54	38.29 ± 2.33
Early-stage-diabetes-data-upload	78.44 ± 2.75	31.47 ± 7.34	11.59 ± 5.57	4.37 ± 4.61	12.18 ± 8.77	10.97 ± 8.57	9.75 ± 7.94	8.41 ± 4.89	9.15 ± 6.25	37.37 ± 9.14
Echocardiogram	89.70 ± 4.98	50.08 ± 25.33	54.44 ± 22.21	59.82 ± 30.06	41.36 ± 18.41	43.30 ± 25.02	43.30 ± 25.02	54.38 ± 28.56	51.03 ± 28.13	32.62 ± 14.70
Ecoli	75.32 ± 1.74	23.54 ± 5.79	27.01 ± 9.08	29.22 ± 8.50	18.34 ± 6.25	18.34 ± 5.94	17.52 ± 4.81	26.48 ± 3.88	26.48 ± 4.78	45.16 ± 9.41
Eegeyesate	91.56 ± 0.57	107.84 ± 4.22	34.17 ± 1.96	33.05 ± 4.11	90.71 ± 0.06	90.63 ± 0.13	90.63 ± 0.13	98.31 ± 1.76	97.85 ± 2.01	75.59 ± 2.38
Electrical	68.31 ± 5.99	13.75 ± 5.86	0.00 ± 0.00	19.92 ± 16.10	38.93 ± 16.77	38.96 ± 14.64	36.80 ± 13.94	4.36 ± 3.42	3.90 ± 3.09	0.73 ± 2.31
Energy-y1	59.86 ± 1.20	30.85 ± 3.38	5.99 ± 2.46	39.08 ± 6.40	18.15 ± 2.23	16.28 ± 2.57	16.28 ± 4.49	20.68 ± 1.21	21.30 ± 2.18	24.83 ± 3.02
Energy-y2	55.82 ± 1.10	26.84 ± 3.05	20.89 ± 3.06	38.15 ± 5.31	16.43 ± 3.27	14.77 ± 2.47	13.94 ± 2.39	19.92 ± 2.78	18.35 ± 3.67	18.52 ± 4.92
Extentionofz-alizadehsani	93.03 ± 1.21	19.68 ± 9.03	1.59 ± 2.20	23.91 ± 12.40	69.98 ± 2.67	69.98 ± 2.67	69.98 ± 2.67	5.18 ± 5.21	4.99 ± 5.25	32.95 ± 11.46
Fertility	96.97 ± 4.64	92.36 ± 13.73	103.29 ± 16.87	80.50 ± 51.66	53.87 ± 10.40	48.98 ± 20.03	48.98 ± 30.55	57.42 ± 28.54	60.18 ± 42.97	53.87 ± 10.40
First-order	97.19 ± 1.01	393.57 ± 21.98	0.00 ± 0.00	3.81 ± 2.79	55.11 ± 1.16	26.10 ± 8.60	11.74 ± 6.33	1.72 ± 1.86	1.44 ± 1.83	53.29 ± 10.21
Flags	92.10 ± 1.78	66.51 ± 7.33	55.39 ± 10.35	70.24 ± 10.52	59.01 ± 11.69	56.97 ± 12.75	58.21 ± 16.34	69.74 ± 13.52	65.32 ± 12.69	53.20 ± 8.94
Foresttypes	61.01 ± 2.62	6.73 ± 5.55	8.70 ± 7.50	8.20 ± 4.86	93.73 ± 3.22	91.00 ± 4.76	91.00 ± 4.76	6.85 ± 4.62	6.50 ± 4.84	23.86 ± 11.45
Garments-worker-productivity	94.96 ± 0.57	97.33 ± 2.17	73.76 ± 5.31	97.73 ± 4.55	85.27 ± 3.86	84.50 ± 3.89	83.51 ± 3.30	86.04 ± 3.49	87.19 ± 3.55	89.36 ± 4.91
Gender-name-dataset	99.13 ± 0.82	86.75 ± 9.13	100.02 ± 0.10	86.20 ± 3.04	74.81 ± 2.89	74.62 ± 4.21	74.49 ± 3.78	100.20 ± 2.29	99.95 ± 2.89	82.34 ± 2.17
Gesture-a1-raw	43.80 ± 1.19	41.31 ± 3.68	13.74 ± 2.79	5.76 ± 1.65	88.48 ± 0.37	88.48 ± 0.37	88.48 ± 0.37	16.33 ± 2.58	15.27 ± 1.27	4.81 ± 1.98
Gesture-a1-va3	76.61 ± 2.00	60.43 ± 4.81	52.58 ± 6.74	38.36 ± 4.03	88.64 ± 0.21	88.64 ± 0.21	88.64 ± 0.21	51.35 ± 3.53	50.57 ± 3.20	56.61 ± 2.98
Gesture-a2-raw	44.81 ± 0.99	45.32 ± 3.76	16.78 ± 3.88	8.14 ± 4.32	87.07 ± 0.36	87.07 ± 0.36	87.07 ± 0.36	18.17 ± 2.42	16.61 ± 2.97	7.56 ± 3.75
Gesture-a2-va3	81.75 ± 2.12	82.33 ± 10.29	63.73 ± 3.73	46.37 ± 5.87	87.27 ± 0.34	87.27 ± 0.34	87.27 ± 0.34	64.21 ± 4.71	63.64 ± 5.66	65.65 ± 5.84
Gesture-a3-raw	34.69 ± 1.28	53.80 ± 3.19	10.48 ± 1.45	5.31 ± 1.33	86.11 ± 0.26	86.11 ± 0.26	86.11 ± 0.26	17.87 ± 1.71	18.09 ± 2.40	6.32 ± 1.56
Gesture-a3-va3	81.14 ± 0.83	67.45 ± 3.72	34.68 ± 4.42	13.90 ± 2.38	86.25 ± 0.28	86.25 ± 0.28	86.25 ± 0.28	57.22 ± 3.36	55.36 ± 3.35	66.08 ± 3.20
Gesture-b1-raw	41.41 ± 2.23	72.03 ± 4.67	11.04 ± 3.44	7.69 ± 3.91	83.92 ± 0.90	83.66 ± 1.05	83.66 ± 1.05	21.63 ± 3.76	17.96 ± 2.56	11.74 ± 3.07
Gesture-b1-va3	83.39 ± 2.35	90.45 ± 8.66	43.56 ± 5.25	13.47 ± 3.67	84.39 ± 0.47	84.39 ± 0.47	84.39 ± 0.47	60.78 ± 3.92	56.72 ± 5.54	78.87 ± 5.95
Gesture-b3-raw	48.32 ± 2.62	60.30 ± 3.59	12.84 ± 4.51	6.27 ± 1.15	87.61 ± 0.43	87.61 ± 0.43	87.61 ± 0.43	18.99 ± 4.84	18.83 ± 4.55	6.07 ± 3.08
Gesture-b3-va3	90.20 ± 1.63	75.18 ± 4.51	57.61 ± 4.64	29.08 ± 3.82	87.73 ± 0.25	87.73 ± 0.25	87.73 ± 0.25	60.88 ± 4.90	62.11 ± 4.47	78.51 ± 7.19
Gesture-c1-raw	47.07 ± 2.15	47.52 ± 3.53	14.73 ± 2.89	7.66 ± 2.13	94.19 ± 0.53	94.19 ± 0.53	94.19 ± 0.53	19.08 ± 3.35	19.07 ± 4.29	7.88 ± 3.30
Gesture-c1-va3	83.85 ± 2.07	62.77 ± 3.70	54.69 ± 4.34	37.89 ± 4.35	94.50 ± 0.42	94.50 ± 0.42	94.50 ± 0.42	50.38 ± 4.02	51.14 ± 3.08	74.23 ± 5.90
Gesture-c3-raw	44.71 ± 1.72	51.60 ± 5.30	16.74 ± 4.75	9.32 ± 2.97	93.71 ± 0.28	93.71 ± 0.28	93.71 ± 0.28	21.17 ± 2.88	23.33 ± 2.29	7.27 ± 1.72

Table 10: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Gesture-c3-va3	86.27 ± 1.65	69.93 ± 2.91	59.86 ± 5.09	48.48 ± 6.04	93.77 ± 0.25	93.77 ± 0.25	93.77 ± 0.25	62.21 ± 4.70	62.77 ± 5.26	73.69 ± 3.58
Glass	74.97 ± 3.67	71.06 ± 9.08	48.23 ± 10.02	42.06 ± 11.80	36.66 ± 8.94	37.99 ± 11.64	39.26 ± 11.50	53.34 ± 9.99	50.21 ± 5.68	58.02 ± 12.16
Go-track-tracks	67.60 ± 8.74	37.42 ± 9.95	27.13 ± 13.92	19.34 ± 16.96	56.18 ± 25.50	56.25 ± 23.12	56.25 ± 23.12	37.59 ± 11.74	36.30 ± 10.90	49.11 ± 27.29
Haberman-survival	94.64 ± 3.43	85.03 ± 6.71	93.09 ± 6.02	86.06 ± 18.01	70.33 ± 11.67	73.69 ± 7.19	73.69 ± 7.19	90.85 ± 6.56	90.66 ± 7.17	69.48 ± 7.26
Hayes-roth	62.76 ± 4.25	67.68 ± 4.69	30.93 ± 7.92	21.33 ± 8.68	21.04 ± 10.92	29.15 ± 10.03	30.33 ± 10.01	51.32 ± 15.37	49.42 ± 15.86	86.18 ± 9.07
Hcc-data	94.91 ± 2.00	70.55 ± 13.41	90.97 ± 11.91	77.33 ± 16.70	80.73 ± 3.96	80.73 ± 3.96	80.73 ± 3.96	75.16 ± 23.89	76.59 ± 20.02	65.20 ± 23.22
Hcvdat	80.27 ± 2.16	38.07 ± 12.46	33.72 ± 8.25	40.46 ± 5.94	53.59 ± 1.50	53.59 ± 1.50	53.59 ± 1.50	32.43 ± 10.46	34.86 ± 10.24	45.68 ± 6.81
Heart-cleveland	91.33 ± 2.49	71.33 ± 10.43	80.13 ± 9.05	70.80 ± 9.79	65.53 ± 12.65	63.49 ± 13.59	65.48 ± 14.24	72.20 ± 8.53	71.26 ± 7.75	73.19 ± 10.59
Heart-hungarian	78.79 ± 4.96	41.67 ± 13.98	59.37 ± 11.70	49.09 ± 17.48	39.56 ± 16.42	38.16 ± 13.73	36.72 ± 13.99	50.25 ± 16.24	50.26 ± 14.06	45.51 ± 20.32
Heart-switzerland	97.90 ± 3.28	92.11 ± 8.88	90.12 ± 15.16	94.39 ± 18.58	84.83 ± 8.69	83.77 ± 7.91	81.35 ± 9.02	84.47 ± 15.43	80.93 ± 13.60	94.32 ± 17.44
Heart-va	97.78 ± 1.25	91.24 ± 7.37	91.84 ± 6.28	85.06 ± 9.93	85.18 ± 12.46	88.43 ± 11.16	91.04 ± 12.31	93.90 ± 7.85	94.42 ± 10.22	94.29 ± 12.81
Heart-failure-clinical-records-dataset	93.16 ± 1.63	57.66 ± 10.11	54.22 ± 10.92	79.25 ± 15.41	73.52 ± 2.86	73.52 ± 2.86	73.52 ± 2.86	62.49 ± 10.67	54.54 ± 18.04	32.91 ± 14.29
Hepatitis	92.08 ± 3.55	54.48 ± 24.83	73.05 ± 20.64	54.33 ± 16.47	50.89 ± 18.80	47.08 ± 28.08	47.08 ± 28.08	61.88 ± 27.40	65.42 ± 34.33	78.56 ± 21.85
Hill-valley	100.16 ± 2.29	102.38 ± 4.86	100.08 ± 0.25	96.41 ± 6.94	95.70 ± 13.15	95.71 ± 14.88	93.73 ± 13.99	91.27 ± 4.53	90.95 ± 4.77	105.01 ± 9.86
Hiv1625data	49.77 ± 3.56	20.84 ± 5.20	36.62 ± 6.44	32.25 ± 3.53	22.02 ± 5.66	18.55 ± 6.01	17.69 ± 6.59	14.33 ± 5.04	18.96 ± 6.49	58.22 ± 4.93
Hiv746data	49.45 ± 3.09	21.20 ± 5.60	51.28 ± 6.19	33.32 ± 6.09	21.02 ± 7.46	18.32 ± 5.62	18.05 ± 6.56	15.19 ± 4.47	16.16 ± 6.28	38.28 ± 8.74
Horse-colic	87.65 ± 2.40	59.11 ± 12.16	33.81 ± 9.25	50.78 ± 16.06	35.25 ± 10.36	33.09 ± 11.30	32.37 ± 12.29	49.23 ± 10.99	47.76 ± 14.44	38.15 ± 11.72
Htru	37.39 ± 2.04	33.26 ± 2.70	20.41 ± 2.69	17.19 ± 2.29	53.21 ± 0.84	51.20 ± 1.14	51.77 ± 1.09	18.13 ± 2.96	17.91 ± 2.34	14.44 ± 2.41
Hypothyroid	93.55 ± 0.93	48.91 ± 5.21	4.16 ± 1.75	58.72 ± 10.61	50.73 ± 2.01	49.82 ± 4.91	50.00 ± 5.21	52.30 ± 32.65	56.91 ± 36.33	25.81 ± 6.37
Ibeacon-rssi-labeled	97.01 ± 0.23	83.22 ± 1.70	75.07 ± 2.27	64.94 ± 2.84	66.64 ± 1.72	64.22 ± 2.84	63.01 ± 3.17	84.54 ± 1.89	82.78 ± 2.15	96.65 ± 1.75
Ilpd-indian-liver	94.61 ± 3.03	108.16 ± 11.80	94.49 ± 8.25	86.13 ± 14.60	70.00 ± 1.38	71.70 ± 4.24	71.29 ± 6.12	84.04 ± 6.74	83.75 ± 7.20	83.01 ± 7.53
Image-segmentation	59.74 ± 3.02	26.23 ± 6.59	15.54 ± 9.18	18.04 ± 5.08	15.00 ± 3.75	15.00 ± 5.27	15.56 ± 5.11	15.24 ± 4.33	14.83 ± 4.43	47.78 ± 10.21
Immunotherapy	89.60 ± 4.21	92.94 ± 19.96	69.14 ± 23.77	91.35 ± 50.14	62.20 ± 8.35	62.20 ± 8.35	62.20 ± 8.35	57.96 ± 25.66	61.26 ± 26.96	45.99 ± 27.09
Impensdata	77.82 ± 2.69	47.91 ± 6.07	99.79 ± 0.00	58.34 ± 8.29	59.21 ± 0.74	59.21 ± 0.74	59.21 ± 0.74	38.09 ± 7.74	43.16 ± 10.61	58.80 ± 7.71
In-vehicle-coupon-recommendation	97.77 ± 0.12	83.15 ± 1.27	76.82 ± 1.34	73.01 ± 2.69	63.41 ± 2.68	62.33 ± 2.67	61.48 ± 2.69	60.94 ± 1.98	59.27 ± 2.63	80.46 ± 1.80
Indian	94.02 ± 2.54	107.75 ± 12.26	80.09 ± 10.71	87.84 ± 16.73	67.51 ± 3.81	69.19 ± 3.32	69.19 ± 3.32	84.54 ± 7.55	83.72 ± 7.19	83.42 ± 7.37
Ionosphere	51.47 ± 4.53	37.70 ± 10.11	20.34 ± 7.16	30.12 ± 9.79	12.34 ± 4.95	11.09 ± 6.88	9.24 ± 6.53	21.13 ± 3.91	20.94 ± 5.71	41.49 ± 16.32
Iris	37.75 ± 5.36	10.00 ± 5.76	7.87 ± 7.85	8.98 ± 8.04	5.00 ± 5.27	6.00 ± 6.99	5.00 ± 5.27	7.36 ± 6.20	6.64 ± 5.32	12.00 ± 9.19
Jain	21.84 ± 8.21	17.35 ± 7.28	1.39 ± 4.38	0.77 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	17.70 ± 6.28	17.64 ± 6.43	12.52 ± 8.66
Jsbach-chorals-harmony	80.38 ± 0.77	22.07 ± 2.53	26.32 ± 2.96	13.94 ± 2.18	18.19 ± 2.95	15.02 ± 2.18	13.64 ± 3.32	12.76 ± 2.82	14.35 ± 4.55	19.65 ± 3.00
Knowledge	76.90 ± 4.13	35.94 ± 11.06	11.39 ± 7.67	36.84 ± 11.45	22.75 ± 13.76	15.49 ± 12.38	11.42 ± 11.02	14.79 ± 6.92	14.60 ± 7.07	24.50 ± 13.45
Lasvegastripadvisorreviews	99.97 ± 0.25	98.75 ± 2.60	97.89 ± 4.52	98.63 ± 4.57	101.19 ± 3.93	103.28 ± 5.62	100.49 ± 5.64	99.59 ± 5.39	98.21 ± 4.86	100.23 ± 3.81
Leaf	82.76 ± 1.37	28.96 ± 7.37	42.06 ± 7.51	100.12 ± 3.03	85.83 ± 4.24	78.52 ± 5.33	77.00 ± 5.75	56.53 ± 5.59	57.23 ± 5.43	78.83 ± 5.06
Led-display	86.13 ± 0.43	42.75 ± 2.23	46.22 ± 2.64	42.55 ± 2.56	31.24 ± 4.55	31.46 ± 4.23	31.80 ± 4.07	42.19 ± 2.22	40.90 ± 2.41	89.39 ± 1.40
Lenses	78.24 ± 19.64	102.13 ± 12.97	43.55 ± 64.20	37.48 ± 34.78	36.36 ± 49.65	32.73 ± 47.28	32.73 ± 47.28	53.43 ± 45.66	55.71 ± 48.24	62.42 ± 68.75
Letter	89.09 ± 0.12	43.64 ± 0.59	14.28 ± 0.71	4.39 ± 0.31	5.33 ± 0.26	4.16 ± 0.31	3.63 ± 0.22	21.06 ± 1.04	20.96 ± 0.80	86.06 ± 0.38
Libras	79.37 ± 2.43	39.44 ± 9.81	33.97 ± 10.24	18.90 ± 5.10	20.23 ± 4.39	15.77 ± 4.66	13.98 ± 4.22	25.08 ± 4.49	23.87 ± 4.32	84.50 ± 6.46
Low-res-spect	74.95 ± 1.64	29.64 ± 5.91	27.01 ± 5.08	26.51 ± 5.94	15.95 ± 3.69	15.66 ± 4.15	14.24 ± 3.85	14.10 ± 4.78	14.10 ± 4.52	39.53 ± 7.99
Lung-cancer	92.89 ± 5.02	56.37 ± 24.31	87.48 ± 36.43	76.93 ± 50.20	70.31 ± 32.97	61.64 ± 32.47	73.12 ± 38.09	79.89 ± 33.05	79.31 ± 34.22	79.46 ± 27.00
Lymphography	84.63 ± 2.26	38.02 ± 11.23	52.62 ± 13.94	48.40 ± 24.02	26.31 ± 17.07	27.84 ± 18.24	31.39 ± 16.28	37.80 ± 14.21	37.20 ± 13.19	45.35 ± 14.88
Magic	74.32 ± 1.89	60.02 ± 4.33	48.72 ± 3.98	44.28 ± 5.09	33.12 ± 4.66	32.55 ± 3.78	32.55 ± 3.08	41.43 ± 4.15	42.65 ± 4.22	61.10 ± 9.66
Mammographic	72.02 ± 2.56	48.19 ± 5.74	49.33 ± 3.99	51.58 ± 9.61	34.73 ± 4.29	35.56 ± 5.22	35.15 ± 5.31	50.99 ± 3.11	52.18 ± 5.22	36.41 ± 2.52
Miniboone	77.60 ± 0.71	180.29 ± 3.52	36.81 ± 6.06	42.03 ± 4.01	39.34 ± 6.27	36.60 ± 5.36	34.65 ± 4.70	53.19 ± 7.06	54.03 ± 8.21	46.39 ± 4.51
Molec-biol-promoter	89.02 ± 4.01	26.91 ± 16.74	53.25 ± 20.10	55.99 ± 23.67	34.72 ± 26.91	30.72 ± 23.89	30.72 ± 23.89	46.07 ± 17.97	46.21 ± 18.16	60.51 ± 22.18
Molec-biol-splice	96.55 ± 0.23	16.95 ± 2.43	13.87 ± 2.87	60.03 ± 3.40	20.95 ± 3.21	20.69 ± 2.78	21.56 ± 2.12	26.61 ± 3.45	27.08 ± 2.93	59.63 ± 3.71

Table 10: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Monks-1	78.06 ± 6.44	73.39 ± 15.31	8.32 ± 13.87	61.26 ± 18.44	38.71 ± 24.21	33.84 ± 21.84	33.71 ± 19.18	27.39 ± 27.93	23.27 ± 32.81	53.45 ± 19.63
Monks-2	100.45 ± 2.64	101.79 ± 5.62	57.45 ± 20.60	82.85 ± 22.09	89.06 ± 15.27	82.81 ± 26.21	75.20 ± 25.61	61.57 ± 19.79	57.60 ± 17.00	87.86 ± 9.58
Monks-3	74.97 ± 6.43	53.25 ± 8.94	21.44 ± 15.42	50.96 ± 30.89	19.62 ± 15.12	21.41 ± 15.91	21.41 ± 15.91	33.73 ± 20.12	25.08 ± 12.86	44.24 ± 7.68
Mushroom	65.43 ± 0.94	24.96 ± 2.21	0.00 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.14 ± 0.01	0.09 ± 0.01	2.96 ± 0.96
Musk-1	80.80 ± 2.69	52.32 ± 14.54	31.55 ± 13.33	31.05 ± 9.22	18.77 ± 8.74	15.39 ± 5.82	12.83 ± 6.08	14.69 ± 4.89	14.71 ± 4.77	76.50 ± 18.17
Musk-2	76.64 ± 1.26	55.96 ± 5.83	18.49 ± 4.90	14.48 ± 3.61	14.72 ± 4.51	9.70 ± 3.58	7.30 ± 2.85	5.64 ± 2.64	5.43 ± 3.23	38.67 ± 4.73
Newdiagnosis	48.01 ± 3.55	20.34 ± 6.58	0.00 ± 0.00	0.17 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.41 ± 0.18	1.02 ± 0.10	46.26 ± 14.11
Nursery	88.77 ± 0.15	31.12 ± 0.76	0.89 ± 0.39	17.22 ± 0.86	2.58 ± 0.32	1.94 ± 0.30	1.56 ± 0.29	7.45 ± 1.96	9.42 ± 2.55	42.53 ± 1.51
Obesitydataset-raw-and-data-sinthetic	78.81 ± 0.39	44.47 ± 1.69	8.71 ± 2.07	21.26 ± 2.88	12.01 ± 2.72	9.68 ± 2.62	8.80 ± 2.59	8.47 ± 3.07	8.33 ± 1.77	38.45 ± 4.18
Obs-network-dataset-2-aug27	45.55 ± 1.79	41.61 ± 7.45	0.52 ± 1.64	1.51 ± 0.92	3.45 ± 0.97	3.45 ± 0.97	3.45 ± 0.97	7.92 ± 2.39	5.39 ± 1.83	13.83 ± 3.08
Occupancy-data	18.67 ± 2.03	10.09 ± 2.20	3.99 ± 1.68	1.95 ± 1.08	41.13 ± 4.35	38.62 ± 4.99	38.62 ± 4.99	7.57 ± 1.99	7.12 ± 2.18	2.83 ± 1.67
Occupancy-data2	22.14 ± 1.37	11.92 ± 1.51	2.71 ± 0.55	1.64 ± 0.68	49.55 ± 1.50	43.21 ± 1.76	43.21 ± 1.76	2.78 ± 0.62	2.62 ± 0.53	1.88 ± 0.47
Occupancy-data3	14.30 ± 1.40	6.64 ± 1.54	2.00 ± 0.45	1.66 ± 0.67	51.10 ± 1.59	46.29 ± 1.83	46.25 ± 1.81	4.03 ± 0.84	3.97 ± 0.88	1.87 ± 0.53
Old	84.69 ± 0.53	15.52 ± 2.13	13.01 ± 2.86	5.28 ± 1.68	9.97 ± 1.88	9.73 ± 2.58	9.15 ± 2.41	6.60 ± 2.04	6.64 ± 2.48	22.91 ± 3.27
Online-shoppers-intention	92.63 ± 0.42	86.25 ± 4.57	53.53 ± 1.74	70.60 ± 2.19	59.24 ± 0.21	59.42 ± 0.49	59.64 ± 0.59	51.02 ± 5.28	52.07 ± 6.12	44.98 ± 3.77
Oocytes-merluccius-nucleus-4d	89.08 ± 2.38	90.22 ± 8.87	65.21 ± 8.13	62.94 ± 8.77	51.35 ± 6.60	44.93 ± 5.64	42.93 ± 6.77	45.77 ± 5.96	47.91 ± 7.66	75.00 ± 11.58
Oocytes-merluccius-states-2f	53.89 ± 4.04	32.92 ± 8.01	23.11 ± 7.17	19.73 ± 6.14	18.04 ± 5.61	17.19 ± 4.55	15.71 ± 4.28	19.18 ± 3.56	19.41 ± 4.66	39.27 ± 10.17
Oocytes-trisopterus-nucleus-2f	89.86 ± 2.47	94.11 ± 10.06	61.67 ± 11.17	52.72 ± 11.86	38.64 ± 7.51	33.25 ± 6.97	34.15 ± 5.66	37.00 ± 5.56	39.20 ± 6.51	83.14 ± 12.22
Oocytes-trisopterus-states-5b	61.02 ± 4.44	48.06 ± 8.10	26.01 ± 6.06	18.56 ± 7.37	16.39 ± 6.78	15.30 ± 5.46	14.42 ± 4.63	14.42 ± 5.03	14.44 ± 5.16	39.78 ± 5.50
Optdigits	89.32 ± 0.15	9.66 ± 1.12	11.34 ± 1.61	1.74 ± 0.71	29.85 ± 2.12	27.92 ± 2.31	27.92 ± 2.31	2.91 ± 0.67	2.43 ± 0.57	81.08 ± 1.37
Optical	88.77 ± 0.22	9.36 ± 1.35	12.64 ± 1.31	2.00 ± 0.63	1.57 ± 0.62	1.39 ± 0.67	1.42 ± 0.65	2.91 ± 0.68	2.93 ± 0.75	80.33 ± 1.44
Ozone	98.65 ± 0.75	514.31 ± 55.69	83.66 ± 10.30	87.49 ± 11.32	51.08 ± 1.86	51.08 ± 1.86	51.08 ± 1.86	67.47 ± 15.39	61.98 ± 13.01	55.98 ± 7.57
Page-blocks	67.45 ± 1.99	57.06 ± 7.90	21.18 ± 2.32	21.68 ± 3.22	20.05 ± 3.93	18.71 ± 3.95	18.52 ± 4.00	26.28 ± 3.56	25.37 ± 3.44	32.81 ± 3.16
Parkingbirmingham	21.04 ± 1.66	31.59 ± 1.11	0.00 ± 0.00	3.44 ± 0.69	42.84 ± 3.92	40.14 ± 4.05	40.14 ± 4.05	27.47 ± 1.45	23.68 ± 1.49	0.00 ± 0.00
Parkinsons	74.34 ± 7.36	82.45 ± 30.00	53.50 ± 10.07	11.04 ± 6.57	32.95 ± 16.92	27.47 ± 16.79	23.27 ± 15.50	31.68 ± 15.10	29.52 ± 19.55	37.10 ± 20.47
Pasture	81.94 ± 8.65	40.75 ± 34.25	35.21 ± 28.56	47.07 ± 37.59	107.02 ± 6.04	107.02 ± 6.04	107.02 ± 6.04	41.92 ± 35.99	43.88 ± 36.49	54.68 ± 39.33
Pbc	83.92 ± 3.21	50.57 ± 13.52	60.78 ± 11.94	85.44 ± 18.48	81.30 ± 1.38	81.30 ± 1.38	81.30 ± 1.38	61.73 ± 10.61	56.72 ± 6.29	58.61 ± 19.36
Pen	68.81 ± 0.36	16.90 ± 1.26	4.44 ± 0.53	0.81 ± 0.19	96.14 ± 0.95	95.06 ± 1.06	95.06 ± 1.06	6.74 ± 0.77	6.83 ± 0.43	67.85 ± 1.71
Pendigits	68.16 ± 0.40	14.41 ± 1.41	4.97 ± 0.79	0.78 ± 0.26	0.43 ± 0.21	0.37 ± 0.17	0.39 ± 0.19	6.10 ± 0.69	6.03 ± 0.60	66.97 ± 2.20
Pharynx	100.44 ± 3.94	107.48 ± 11.56	99.56 ± 0.04	100.09 ± 29.37	67.52 ± 22.16	73.58 ± 28.05	81.86 ± 28.07	80.91 ± 10.33	82.63 ± 43.90	201.57 ± 18.02
Phishingwebsites	87.61 ± 0.21	18.12 ± 1.15	11.49 ± 0.83	6.55 ± 1.11	11.14 ± 1.05	10.14 ± 1.02	9.60 ± 0.93	6.99 ± 0.83	6.88 ± 0.63	22.51 ± 1.14
Pima	86.45 ± 3.14	62.48 ± 7.65	69.45 ± 9.95	65.77 ± 10.36	53.56 ± 8.94	54.71 ± 8.81	55.00 ± 7.52	65.00 ± 5.29	63.83 ± 4.80	62.74 ± 11.34
Pittsburg-bridges-rel-l	85.63 ± 4.14	66.08 ± 15.18	76.14 ± 17.01	49.78 ± 27.10	54.48 ± 20.28	52.97 ± 20.36	45.14 ± 18.33	67.73 ± 16.66	64.05 ± 17.55	48.20 ± 17.58
Pittsburg-bridges-span	86.15 ± 3.47	64.47 ± 11.41	72.83 ± 17.00	72.74 ± 26.22	56.74 ± 24.06	58.65 ± 26.75	58.65 ± 26.75	58.68 ± 11.30	54.18 ± 17.94	73.56 ± 22.28
Pittsburg-bridges-t-or-d	92.43 ± 8.61	80.50 ± 21.12	86.75 ± 31.08	69.10 ± 43.95	53.74 ± 26.20	47.70 ± 33.01	43.14 ± 29.26	58.96 ± 44.70	65.23 ± 51.65	60.71 ± 42.49
Pittsburg-bridges-type	89.03 ± 3.97	67.18 ± 9.43	65.56 ± 7.28	58.61 ± 14.73	60.67 ± 16.06	50.95 ± 23.20	49.46 ± 17.71	60.92 ± 18.91	59.26 ± 20.97	57.79 ± 7.39
Pittsburg-bridgesmaterial	73.99 ± 3.79	45.14 ± 15.62	47.53 ± 13.35	42.97 ± 27.99	33.41 ± 14.32	35.53 ± 13.89	38.09 ± 22.43	45.06 ± 20.52	52.34 ± 19.88	31.28 ± 10.34
Planning	99.62 ± 2.38	104.13 ± 8.76	99.73 ± 0.02	83.73 ± 22.67	69.71 ± 2.82	72.56 ± 14.76	83.38 ± 17.31	101.14 ± 12.75	105.45 ± 20.71	92.84 ± 21.47
Plant-margin	94.95 ± 0.12	15.73 ± 2.46	54.50 ± 4.17	31.05 ± 2.63	17.36 ± 2.29	15.40 ± 2.83	15.02 ± 2.85	24.09 ± 1.81	23.12 ± 2.51	93.24 ± 2.19
Plant-shape	91.94 ± 0.11	47.32 ± 3.03	55.79 ± 3.25	39.84 ± 3.16	51.57 ± 1.63	43.68 ± 3.16	40.34 ± 3.39	43.86 ± 3.22	41.23 ± 3.03	92.35 ± 1.63
Plant-texture	95.39 ± 0.15	25.87 ± 3.18	50.29 ± 3.39	25.28 ± 2.68	16.86 ± 2.52	14.59 ± 2.31	14.08 ± 2.37	23.84 ± 1.60	22.80 ± 1.59	97.53 ± 0.91
Poker-hand-training-true	99.97 ± 0.01	99.98 ± 0.03	84.40 ± 2.91	93.48 ± 1.51	72.66 ± 1.76	74.66 ± 1.64	76.64 ± 1.96	96.31 ± 1.26	96.86 ± 1.79	88.07 ± 0.03
Post-operative	97.54 ± 2.88	103.29 ± 8.23	97.97 ± 0.37	101.82 ± 20.36	66.34 ± 9.65	71.48 ± 12.99	76.48 ± 17.86	97.20 ± 19.90	101.48 ± 28.67	71.48 ± 12.99
Primary-tumor	95.28 ± 0.69	68.67 ± 6.71	73.20 ± 6.01	76.00 ± 9.25	60.18 ± 12.24	62.24 ± 11.86	63.26 ± 13.42	70.17 ± 7.30	70.44 ± 7.80	81.40 ± 3.14
Qsarbioconcentration	100.17 ± 0.61	100.83 ± 4.02	102.60 ± 5.96	102.26 ± 15.18	66.97 ± 1.48	71.09 ± 4.12	78.25 ± 7.59	104.51 ± 6.05	104.62 ± 4.85	72.41 ± 8.44

Table 10: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Qsarbiodegradation	88.05 ± 1.73	53.37 ± 6.92	42.98 ± 6.42	36.83 ± 5.01	31.57 ± 5.87	30.73 ± 4.85	32.22 ± 5.57	32.16 ± 3.76	32.35 ± 3.67	52.10 ± 8.81
Qualitative-bankruptcy	26.63 ± 1.92	1.79 ± 3.29	4.80 ± 6.53	0.66 ± 0.54	2.46 ± 5.54	2.46 ± 5.54	0.81 ± 2.57	1.60 ± 3.70	1.48 ± 3.65	3.28 ± 5.74
Ringnorm	59.88 ± 2.39	3.92 ± 0.66	18.08 ± 0.37	49.70 ± 3.17	2.81 ± 0.83	3.05 ± 0.82	3.30 ± 0.97	16.74 ± 2.49	17.42 ± 1.69	71.14 ± 1.93
Risk-factors-cervical-cancer	96.16 ± 1.09	92.97 ± 11.30	49.02 ± 12.44	47.55 ± 18.02	52.88 ± 2.93	53.88 ± 5.59	49.80 ± 12.09	41.47 ± 12.51	45.32 ± 16.32	31.89 ± 16.03
Robotnavigation	47.91 ± 1.23	71.45 ± 2.17	0.69 ± 0.37	17.81 ± 1.83	15.05 ± 1.73	13.31 ± 1.48	12.09 ± 1.17	19.84 ± 2.42	19.17 ± 3.47	37.18 ± 2.86
Sapfile	95.19 ± 1.58	77.07 ± 9.48	92.26 ± 18.29	96.60 ± 13.95	79.45 ± 13.75	78.18 ± 10.63	78.15 ± 13.06	81.94 ± 12.13	82.67 ± 16.59	90.31 ± 19.72
Sat	51.32 ± 0.52	25.36 ± 1.49	18.48 ± 1.13	11.75 ± 0.97	91.15 ± 0.66	89.69 ± 0.58	89.69 ± 0.58	13.80 ± 1.71	14.23 ± 1.26	49.67 ± 1.61
Satelite	51.35 ± 0.60	25.36 ± 1.36	18.21 ± 0.85	11.72 ± 1.37	91.17 ± 0.71	89.59 ± 0.88	89.59 ± 0.88	13.49 ± 0.66	14.00 ± 1.03	49.40 ± 2.03
Scadi	84.82 ± 1.64	22.22 ± 11.10	32.14 ± 15.34	35.37 ± 17.03	32.36 ± 12.76	28.64 ± 18.71	24.89 ± 20.41	28.81 ± 14.85	28.22 ± 14.97	55.28 ± 13.83
Schillingdata	82.03 ± 2.00	42.45 ± 2.63	99.92 ± 0.00	63.68 ± 3.93	57.60 ± 0.40	47.78 ± 4.43	36.36 ± 2.87	32.39 ± 8.29	39.05 ± 12.63	58.26 ± 1.72
Seeds	42.07 ± 5.78	14.04 ± 6.27	14.78 ± 8.74	10.00 ± 2.96	9.29 ± 5.88	9.29 ± 5.88	9.29 ± 5.88	9.26 ± 3.26	8.33 ± 3.11	26.43 ± 15.08
Segment	53.80 ± 1.24	23.49 ± 1.80	4.27 ± 1.12	3.65 ± 0.72	40.40 ± 3.36	38.48 ± 2.80	38.54 ± 2.88	6.42 ± 1.05	5.63 ± 0.83	42.17 ± 3.24
Seismic-bumps	98.69 ± 0.40	115.78 ± 13.12	98.60 ± 3.58	86.32 ± 11.80	53.38 ± 0.06	53.38 ± 0.06	53.38 ± 0.06	78.60 ± 8.88	75.65 ± 8.05	57.15 ± 4.41
Semeion	96.78 ± 0.07	15.63 ± 3.03	29.94 ± 2.41	10.00 ± 1.86	4.88 ± 1.19	4.26 ± 0.77	4.33 ± 0.80	9.99 ± 1.85	9.37 ± 1.92	89.35 ± 0.53
Setapprocesst1	93.44 ± 5.93	96.50 ± 43.94	87.03 ± 39.95	81.33 ± 23.35	83.51 ± 18.75	72.54 ± 24.34	76.34 ± 21.09	82.46 ± 42.80	85.72 ± 44.19	64.52 ± 53.46
Setapprocesst10	98.97 ± 3.13	124.35 ± 32.41	94.02 ± 29.97	88.89 ± 36.54	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	105.91 ± 54.99	105.17 ± 48.88	84.35 ± 17.77
Setapprocesst11	98.56 ± 4.45	101.50 ± 58.38	81.02 ± 38.41	93.61 ± 33.26	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	86.46 ± 33.25	78.38 ± 37.24	95.80 ± 22.70
Setapprocesst2	97.76 ± 3.58	73.36 ± 30.55	59.92 ± 40.59	83.49 ± 46.02	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	84.47 ± 38.01	78.61 ± 38.22	41.47 ± 37.02
Setapprocesst3	97.51 ± 3.17	78.93 ± 26.90	88.97 ± 34.49	76.68 ± 30.30	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	77.26 ± 27.64	77.43 ± 19.43	104.81 ± 36.47
Setapprocesst4	99.87 ± 2.10	122.70 ± 60.90	110.95 ± 35.48	97.13 ± 42.87	68.66 ± 13.00	68.66 ± 13.00	68.66 ± 13.00	109.70 ± 45.59	105.27 ± 38.88	66.22 ± 27.24
Setapprocesst5	96.83 ± 4.53	118.50 ± 38.81	105.16 ± 36.94	90.15 ± 36.20	77.22 ± 14.45	79.94 ± 17.59	79.94 ± 17.59	91.72 ± 26.54	86.41 ± 25.92	77.05 ± 57.09
Setapprocesst6	97.09 ± 2.12	72.69 ± 34.89	78.35 ± 27.56	80.69 ± 31.47	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	82.28 ± 31.96	83.55 ± 31.72	89.05 ± 46.01
Setapprocesst7	97.36 ± 4.02	57.12 ± 30.93	87.54 ± 43.22	80.30 ± 31.09	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	62.70 ± 38.99	64.34 ± 39.59	105.09 ± 34.29
Setapprocesst8	96.38 ± 2.61	70.34 ± 32.11	67.05 ± 33.15	83.06 ± 34.00	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	76.02 ± 35.53	74.51 ± 36.64	113.36 ± 28.85
Setapprocesst9	98.09 ± 2.65	78.22 ± 32.68	70.80 ± 35.69	93.07 ± 20.30	74.50 ± 9.59	74.50 ± 9.59	74.50 ± 9.59	64.42 ± 39.11	73.83 ± 36.60	110.64 ± 39.73
Shillbiddingdataset	64.59 ± 0.97	18.80 ± 1.93	2.83 ± 0.76	4.15 ± 1.48	2.40 ± 0.91	2.32 ± 1.09	1.82 ± 1.02	2.19 ± 0.88	1.92 ± 1.04	14.25 ± 2.80
Shuttle-landing-control	98.09 ± 98.47	153.52 ± 221.31	61.65 ± 13.48	81.09 ± 116.78	10.00 ± 31.62	10.00 ± 31.62	10.00 ± 31.62	94.74 ± 194.49	98.24 ± 206.94	90.00 ± 251.44
Somervillehappinesssurvey2015	93.67 ± 5.90	89.07 ± 11.60	82.68 ± 20.00	83.95 ± 20.26	80.34 ± 21.75	83.02 ± 20.63	82.95 ± 23.46	84.30 ± 21.18	85.77 ± 21.62	69.05 ± 31.84
Sonar	90.81 ± 2.64	62.61 ± 17.91	57.49 ± 14.47	27.74 ± 13.91	68.49 ± 10.87	59.72 ± 18.03	53.03 ± 19.12	38.08 ± 19.26	37.80 ± 18.79	75.34 ± 22.26
Soybean	88.59 ± 0.63	9.44 ± 5.73	15.32 ± 5.03	18.94 ± 4.92	10.70 ± 6.56	8.56 ± 4.87	9.26 ± 5.13	15.90 ± 6.44	15.33 ± 5.95	75.82 ± 2.65
Spambase	84.20 ± 0.69	43.63 ± 3.71	18.72 ± 2.09	19.36 ± 2.26	13.84 ± 1.95	13.61 ± 2.30	13.79 ± 1.93	22.48 ± 2.57	27.47 ± 12.10	45.33 ± 4.18
Speaker-accent	88.94 ± 2.95	65.19 ± 10.75	45.75 ± 13.13	29.60 ± 12.53	59.48 ± 5.54	53.85 ± 6.07	53.85 ± 6.07	36.25 ± 8.70	37.57 ± 9.12	70.82 ± 6.72
Spect	94.05 ± 5.88	70.52 ± 33.64	86.21 ± 21.82	111.23 ± 27.49	79.59 ± 25.42	80.41 ± 27.23	83.14 ± 22.23	105.25 ± 38.65	105.67 ± 38.35	62.49 ± 20.91
Spectf	93.47 ± 4.09	49.03 ± 24.59	56.85 ± 22.77	65.95 ± 36.62	65.00 ± 12.91	65.00 ± 12.91	47.50 ± 27.51	59.54 ± 28.00	57.60 ± 26.15	57.50 ± 16.87
Statlog-australian-credit	98.53 ± 0.69	105.50 ± 4.70	92.82 ± 10.37	101.28 ± 7.68	74.34 ± 4.06	81.65 ± 14.63	82.63 ± 12.80	93.88 ± 7.57	95.03 ± 7.04	79.67 ± 14.60
Statlog-german-credit	97.57 ± 0.32	68.80 ± 7.51	72.76 ± 9.80	76.73 ± 9.35	56.88 ± 9.81	54.02 ± 8.91	54.02 ± 10.53	70.24 ± 8.41	66.81 ± 7.40	69.26 ± 5.43
Statlog-heart	79.99 ± 3.41	37.70 ± 13.10	56.04 ± 16.74	50.66 ± 17.20	34.50 ± 10.72	40.50 ± 17.03	41.25 ± 19.44	45.54 ± 11.90	41.61 ± 15.50	58.49 ± 12.65
Statlog-image	50.97 ± 1.19	23.62 ± 1.87	4.25 ± 1.14	3.65 ± 0.72	6.67 ± 1.03	5.66 ± 1.11	5.30 ± 1.04	6.16 ± 1.00	6.14 ± 1.22	42.37 ± 2.87
Statlog-landsat	50.64 ± 0.88	25.41 ± 2.74	18.67 ± 2.16	12.33 ± 2.10	13.03 ± 1.96	12.13 ± 1.71	12.05 ± 1.99	14.70 ± 1.53	14.52 ± 2.13	48.93 ± 2.08
Statlog-shuttle	72.47 ± 0.54	27.12 ± 1.38	0.14 ± 0.07	0.21 ± 0.10	0.70 ± 0.13	0.57 ± 0.13	0.55 ± 0.12	1.28 ± 0.17	1.26 ± 0.14	14.83 ± 0.84
Statlog-vehicle	77.89 ± 1.57	74.64 ± 3.21	37.89 ± 5.61	40.51 ± 5.94	31.06 ± 5.18	26.33 ± 3.47	26.01 ± 3.15	26.89 ± 3.52	26.91 ± 4.84	65.11 ± 6.80
Steel-plates	74.89 ± 1.28	51.74 ± 4.59	32.78 ± 3.85	36.31 ± 3.49	31.44 ± 3.56	29.59 ± 2.90	29.72 ± 3.74	37.74 ± 2.85	37.67 ± 2.92	66.60 ± 4.99
Synthetic-control	67.81 ± 1.68	6.59 ± 3.88	10.39 ± 4.21	5.25 ± 2.55	0.80 ± 1.03	0.60 ± 0.97	0.60 ± 0.97	2.38 ± 1.17	2.10 ± 1.14	51.60 ± 4.60
Teaching	91.04 ± 3.59	85.30 ± 6.22	65.99 ± 14.73	56.33 ± 19.19	65.76 ± 22.04	57.76 ± 19.01	56.76 ± 18.56	80.76 ± 7.84	80.52 ± 7.84	80.45 ± 16.39
Thoraricsurgery	98.87 ± 1.05	111.80 ± 29.70	99.67 ± 0.22	89.90 ± 23.21	58.48 ± 0.00	60.99 ± 4.04	60.99 ± 4.04	92.81 ± 15.64	86.92 ± 17.56	65.17 ± 6.59
Thyroid	57.77 ± 8.05	7.98 ± 7.66	18.77 ± 13.14	8.02 ± 5.05	51.27 ± 10.32	48.41 ± 9.52	48.41 ± 9.52	9.32 ± 6.57	8.93 ± 6.43	19.81 ± 18.26

Table 10: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Thyroid-train	92.47 ± 1.02	47.14 ± 5.66	2.20 ± 2.26	55.74 ± 9.39	34.50 ± 4.86	30.39 ± 4.22	26.29 ± 4.44	30.70 ± 6.72	35.07 ± 10.96	24.80 ± 5.47
Tic-tac-toe	96.31 ± 0.52	82.35 ± 1.30	15.71 ± 4.65	0.18 ± 0.01	3.00 ± 3.10	1.62 ± 2.20	0.69 ± 1.57	6.34 ± 2.17	5.45 ± 2.62	66.37 ± 5.27
Titanic	83.80 ± 1.82	71.83 ± 3.90	71.99 ± 3.00	70.67 ± 2.81	49.24 ± 5.01	48.20 ± 4.60	48.20 ± 4.60	73.76 ± 4.95	73.97 ± 6.57	51.21 ± 4.63
Trains	84.82 ± 15.40	73.34 ± 94.66	51.94 ± 84.05	76.67 ± 77.46	73.33 ± 94.67	55.00 ± 88.56	55.00 ± 88.56	50.95 ± 67.29	49.70 ± 68.53	183.33 ± 0.00
Transfusion	96.33 ± 1.56	80.83 ± 5.96	83.64 ± 3.41	87.89 ± 11.10	68.12 ± 7.91	74.38 ± 8.43	77.33 ± 10.26	81.45 ± 4.21	81.29 ± 5.46	65.18 ± 4.59
Trial	68.74 ± 3.53	20.60 ± 7.26	0.00 ± 0.00	0.84 ± 1.75	1.92 ± 1.85	1.37 ± 1.45	1.37 ± 1.45	2.16 ± 1.92	1.78 ± 1.94	0.00 ± 0.00
Turkiye-student-evaluation	93.92 ± 0.34	27.54 ± 2.76	0.06 ± 0.13	17.88 ± 2.93	8.37 ± 0.97	6.93 ± 1.10	6.93 ± 1.10	0.20 ± 0.02	0.14 ± 0.02	0.06 ± 0.13
Unbalanced	95.88 ± 1.76	389.07 ± 168.32	95.69 ± 0.55	87.80 ± 59.75	47.45 ± 8.26	56.16 ± 18.15	63.68 ± 22.74	88.33 ± 22.73	82.28 ± 28.37	47.45 ± 8.26
Urbanlandcover	80.42 ± 3.34	23.92 ± 10.79	25.59 ± 7.52	30.66 ± 11.82	93.85 ± 2.26	93.85 ± 2.26	93.85 ± 2.26	27.55 ± 10.22	27.18 ± 10.37	57.61 ± 16.50
Userknowledgemodeling	75.33 ± 5.18	30.86 ± 5.69	11.70 ± 4.63	28.66 ± 14.08	31.26 ± 9.01	20.64 ± 6.01	20.09 ± 5.50	12.98 ± 4.39	11.14 ± 5.47	20.65 ± 11.01
Vehicle	77.45 ± 1.66	75.41 ± 3.51	37.74 ± 5.74	40.51 ± 5.94	92.71 ± 3.38	91.61 ± 4.09	91.61 ± 4.09	27.03 ± 4.00	26.45 ± 5.37	64.16 ± 6.72
Vertebral-column-2classes	78.84 ± 9.07	50.23 ± 15.73	47.01 ± 14.02	41.81 ± 11.55	33.18 ± 12.17	35.39 ± 14.25	34.65 ± 13.03	41.47 ± 9.67	40.32 ± 9.74	59.72 ± 17.53
Vertebral-column-3classes	66.41 ± 4.10	33.87 ± 5.65	32.10 ± 9.04	34.85 ± 7.69	25.81 ± 9.43	24.27 ± 11.17	25.30 ± 10.73	31.61 ± 5.37	30.66 ± 5.10	39.75 ± 11.17
Veteran	92.97 ± 1.93	75.26 ± 12.85	80.96 ± 8.67	95.26 ± 21.20	70.42 ± 1.88	70.42 ± 1.88	70.42 ± 1.88	93.54 ± 29.30	81.84 ± 28.68	63.59 ± 17.95
Vowel	80.68 ± 1.05	51.58 ± 3.76	21.88 ± 5.27	1.99 ± 0.90	16.44 ± 2.86	9.78 ± 2.21	7.78 ± 2.28	12.32 ± 3.37	10.10 ± 2.58	74.78 ± 4.63
Wall-following	48.03 ± 1.15	71.46 ± 1.43	0.60 ± 0.39	17.92 ± 1.83	16.60 ± 2.13	14.19 ± 2.39	13.06 ± 1.96	20.34 ± 2.17	19.91 ± 2.62	36.77 ± 2.15
Waveform-noise	82.77 ± 0.94	30.52 ± 2.17	38.30 ± 1.82	39.61 ± 1.90	20.79 ± 2.27	21.93 ± 1.92	22.65 ± 2.12	24.73 ± 2.35	25.51 ± 2.03	69.42 ± 4.12
Waveform	75.61 ± 0.91	29.23 ± 1.85	38.13 ± 1.95	34.70 ± 3.02	20.37 ± 2.04	21.51 ± 2.07	21.81 ± 2.15	24.51 ± 1.76	25.52 ± 1.87	71.20 ± 2.46
Wbc	21.48 ± 3.02	5.64 ± 3.13	19.31 ± 4.17	10.08 ± 4.61	5.15 ± 3.47	5.15 ± 3.47	4.83 ± 3.81	9.33 ± 3.05	8.55 ± 4.04	19.64 ± 6.41
Wdbc	52.63 ± 4.72	14.94 ± 8.88	15.44 ± 8.53	9.03 ± 3.99	79.67 ± 1.28	79.67 ± 1.28	79.67 ± 1.28	7.46 ± 3.60	8.46 ± 4.45	21.79 ± 6.79
Weathernominal	99.31 ± 34.07	89.36 ± 34.03	79.58 ± 76.74	105.12 ± 59.86	55.00 ± 59.86	80.00 ± 82.33	55.00 ± 83.17	55.18 ± 81.59	55.09 ± 82.12	115.00 ± 97.33
Weathernumeric	117.85 ± 31.20	94.85 ± 27.94	51.25 ± 72.28	53.86 ± 72.31	55.00 ± 59.86	100.00 ± 97.18	100.00 ± 97.18	57.19 ± 82.33	58.08 ± 82.85	115.00 ± 97.33
Website-phishingdata	77.13 ± 1.04	37.13 ± 3.15	22.20 ± 3.01	24.98 ± 3.42	25.55 ± 2.26	24.63 ± 2.13	22.00 ± 3.66	22.88 ± 3.32	21.51 ± 3.94	32.53 ± 2.92
Wholesalecustomersdata	52.14 ± 4.41	27.19 ± 6.34	30.44 ± 6.43	27.98 ± 10.93	73.76 ± 1.66	73.76 ± 1.66	73.76 ± 1.66	24.95 ± 5.71	24.87 ± 5.84	21.82 ± 8.40
Wifi-localization	53.31 ± 0.96	3.08 ± 0.75	4.59 ± 0.68	2.48 ± 1.18	27.93 ± 3.32	26.07 ± 3.43	26.07 ± 3.43	3.69 ± 0.48	3.36 ± 0.70	28.00 ± 5.94
Wilt	98.25 ± 1.77	169.09 ± 9.21	21.81 ± 4.42	50.33 ± 5.73	51.14 ± 1.29	51.14 ± 1.29	51.14 ± 1.29	26.92 ± 4.66	22.69 ± 3.57	54.17 ± 1.88
Wine-quality-red	85.83 ± 2.01	82.11 ± 3.21	63.71 ± 3.92	55.01 ± 4.21	57.83 ± 3.82	57.24 ± 4.11	57.05 ± 4.14	77.23 ± 3.54	75.38 ± 3.21	70.46 ± 4.77
Wine-quality-white	90.26 ± 0.40	89.15 ± 0.96	65.18 ± 3.20	51.32 ± 3.70	63.63 ± 3.38	62.36 ± 3.65	62.00 ± 3.90	82.93 ± 1.44	82.53 ± 1.38	80.28 ± 3.17
Wine	52.96 ± 3.13	5.09 ± 7.02	11.00 ± 7.84	9.36 ± 6.11	1.68 ± 3.55	2.58 ± 4.16	2.58 ± 4.16	5.66 ± 4.98	5.43 ± 5.11	34.99 ± 14.01
Yamilnaduelectricty	92.30 ± 0.69	100.00 ± 0.00	92.08 ± 0.89	83.42 ± 1.38	98.78 ± 0.58	98.61 ± 0.59	98.74 ± 0.64	99.98 ± 0.04	99.98 ± 0.05	97.94 ± 1.18
Yeast	91.42 ± 0.61	67.16 ± 2.46	65.36 ± 4.00	61.75 ± 3.05	51.13 ± 5.26	51.04 ± 5.46	50.87 ± 5.80	65.48 ± 2.40	65.15 ± 2.53	77.12 ± 4.38
Youtobe-kabita-preprocessing	92.32 ± 0.36	86.08 ± 0.85	83.77 ± 1.30	83.17 ± 1.38	73.12 ± 2.76	74.40 ± 2.56	75.21 ± 2.46	87.94 ± 0.65	87.78 ± 0.65	78.90 ± 2.34
Youtobe-nisha-preprocessing	91.64 ± 0.26	85.67 ± 0.84	81.03 ± 0.87	80.28 ± 1.97	69.48 ± 2.48	70.90 ± 2.45	71.38 ± 2.60	86.17 ± 0.67	86.08 ± 0.95	80.24 ± 1.53
Z-alizadehsani	95.90 ± 1.24	48.74 ± 14.11	57.45 ± 18.02	53.44 ± 16.65	69.98 ± 2.67	69.98 ± 2.67	69.98 ± 2.67	42.10 ± 17.85	43.58 ± 15.44	71.56 ± 12.76
Zoo	65.90 ± 3.25	5.23 ± 8.36	10.06 ± 11.09	8.68 ± 6.54	7.54 ± 10.51	7.54 ± 10.51	7.54 ± 10.51	8.97 ± 7.18	7.92 ± 7.07	34.76 ± 13.50
Average rae (rank)	79.08 (10)	60.74 (9)	44.76 (3)	45.24 (4)	49.04 (7)	47.95 (6)	47.48 (5)	43.61 (2)	43.27 (1)	56.74 (8)
Average rae std (rank)	3.21 (1)	10.91 (7)	9.45 (5)	11.66 (9)	7.55 (2)	8.53 (3)	8.75 (4)	10.86 (6)	11.51 (8)	11.75 (10)

Table 11: The detailed experimental results on root relative squared error and standard deviation.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Abalone	87.04 ± 1.45	98.65 ± 3.48	96.11 ± 1.73	112.29 ± 3.33	100.03 ± 3.71	99.50 ± 3.56	99.58 ± 3.38	81.78 ± 1.93	82.19 ± 2.46	108.24 ± 2.48
Absenteeism-at-work	64.13 ± 1.29	65.75 ± 7.93	5.26 ± 9.63	81.19 ± 7.45	78.51 ± 8.03	78.51 ± 8.03	78.51 ± 8.03	29.02 ± 6.25	29.45 ± 8.61	43.21 ± 8.16
Acute-inflammation	49.79 ± 4.91	30.75 ± 6.75	0.00 ± 0.00	1.68 ± 0.08	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.23 ± 0.11	0.89 ± 0.09	89.41 ± 19.34
Acute-nephritis	44.00 ± 2.74	28.44 ± 26.94	0.00 ± 0.00	1.69 ± 0.08	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.23 ± 0.14	0.98 ± 0.16	44.83 ± 39.70
Adult	94.87 ± 0.89	94.07 ± 7.03	88.25 ± 6.56	108.54 ± 8.94	95.11 ± 11.86	94.77 ± 11.64	94.93 ± 11.80	89.31 ± 8.09	90.37 ± 7.98	103.82 ± 3.43
Aggregation	41.47 ± 7.37	5.51 ± 3.22	5.39 ± 8.69	2.73 ± 5.41	1.82 ± 5.75	3.62 ± 7.64	3.62 ± 7.64	16.58 ± 4.55	15.24 ± 6.30	101.51 ± 3.69
Algerianforest	67.39 ± 5.59	38.32 ± 28.13	32.55 ± 26.76	66.75 ± 19.67	129.31 ± 2.58	124.52 ± 6.88	124.52 ± 6.88	29.20 ± 20.27	23.61 ± 23.09	18.48 ± 30.00
Annealing	92.46 ± 0.55	110.20 ± 10.34	37.23 ± 13.89	71.37 ± 8.09	73.64 ± 10.23	64.64 ± 9.73	62.34 ± 8.38	58.54 ± 14.28	57.72 ± 12.39	90.27 ± 0.06
Arrhythmia	97.62 ± 0.51	105.27 ± 9.70	95.99 ± 10.56	115.41 ± 4.38	106.37 ± 3.39	100.94 ± 5.15	102.18 ± 6.11	89.64 ± 8.21	101.99 ± 11.41	109.41 ± 3.31
Au1-1000	99.16 ± 0.18	97.22 ± 2.15	97.74 ± 9.49	116.84 ± 8.88	116.17 ± 0.26	116.17 ± 0.26	115.48 ± 1.85	121.39 ± 7.93	116.95 ± 8.29	116.17 ± 0.26
Au4-2500	95.57 ± 0.69	98.30 ± 2.76	93.63 ± 8.56	113.00 ± 7.26	116.52 ± 4.28	116.52 ± 4.28	116.52 ± 4.28	101.91 ± 5.90	102.40 ± 6.97	127.86 ± 5.57
Au6-1000	100.19 ± 0.35	103.39 ± 0.83	130.38 ± 3.17	141.35 ± 3.10	133.03 ± 0.02	133.03 ± 0.02	133.03 ± 0.02	129.97 ± 2.01	131.94 ± 3.22	132.76 ± 1.94
Au6-250-drift-au6-cd1-500	100.42 ± 0.60	104.58 ± 1.76	130.17 ± 3.00	139.11 ± 3.50	135.09 ± 0.52	135.09 ± 0.52	135.09 ± 0.52	128.72 ± 1.87	128.09 ± 3.59	132.84 ± 3.07
Au6-cd1-400	100.02 ± 0.56	107.06 ± 3.95	116.33 ± 6.32	139.16 ± 6.89	131.99 ± 0.50	131.99 ± 0.50	131.99 ± 0.50	124.44 ± 5.63	125.69 ± 4.62	128.15 ± 6.26
Au7-300-drift-au7-cpd1-800	99.03 ± 0.63	97.68 ± 1.41	114.95 ± 4.82	130.95 ± 4.15	134.79 ± 1.52	135.47 ± 1.81	135.47 ± 1.81	106.21 ± 3.72	106.22 ± 3.32	137.99 ± 2.84
Au7-700	98.90 ± 1.36	98.55 ± 2.70	111.03 ± 4.59	138.48 ± 6.11	137.25 ± 3.13	138.19 ± 2.55	138.19 ± 2.55	113.58 ± 3.03	111.08 ± 3.93	128.82 ± 4.47
Au7-cpd1-500	99.46 ± 0.85	98.56 ± 2.77	102.46 ± 6.33	128.08 ± 5.92	128.48 ± 0.87	128.06 ± 1.63	128.06 ± 1.63	115.19 ± 4.86	115.27 ± 6.42	126.30 ± 5.00
Audiology-std	97.30 ± 0.37	71.56 ± 13.39	66.74 ± 15.49	72.36 ± 17.49	92.29 ± 10.19	85.55 ± 15.05	81.75 ± 12.99	55.26 ± 10.04	58.45 ± 7.30	110.81 ± 2.16
Audit-risk	62.32 ± 3.32	45.95 ± 9.60	2.31 ± 7.32	28.16 ± 11.64	20.84 ± 15.36	17.56 ± 15.99	15.22 ± 16.74	31.35 ± 11.06	32.12 ± 10.84	0.00 ± 0.00
Autism-adolescent-data	79.76 ± 5.66	21.72 ± 14.92	0.00 ± 0.00	55.68 ± 30.30	25.41 ± 32.81	31.66 ± 33.38	25.41 ± 32.81	47.83 ± 37.02	47.60 ± 37.80	0.00 ± 0.00
Autism-adult-data	76.53 ± 2.98	27.93 ± 13.51	0.00 ± 0.00	47.57 ± 19.04	11.90 ± 15.68	6.49 ± 13.93	9.17 ± 15.07	1.85 ± 1.12	1.61 ± 1.08	0.00 ± 0.00
Autism-child-data	81.09 ± 2.35	20.96 ± 7.07	0.00 ± 0.00	66.42 ± 11.51	3.72 ± 11.75	0.00 ± 0.00	0.00 ± 0.00	6.57 ± 7.91	7.57 ± 8.33	0.00 ± 0.00
Autos	79.44 ± 4.00	93.69 ± 15.49	59.87 ± 12.16	74.84 ± 18.73	130.14 ± 4.27	129.13 ± 5.18	129.13 ± 5.18	69.30 ± 10.55	69.54 ± 10.22	98.15 ± 12.45
Avila	55.95 ± 0.61	113.05 ± 2.00	36.89 ± 3.61	79.05 ± 4.37	91.07 ± 3.03	87.94 ± 2.73	85.79 ± 3.46	81.95 ± 1.65	81.87 ± 2.52	85.67 ± 3.79
Balance-scale	78.97 ± 2.17	64.09 ± 1.77	84.72 ± 6.53	74.91 ± 9.18	57.92 ± 4.57	50.23 ± 3.06	47.21 ± 5.01	46.13 ± 7.33	47.42 ± 7.76	123.73 ± 6.26
Balloons	84.28 ± 22.19	100.00 ± 0.00	101.11 ± 32.98	73.52 ± 19.16	73.13 ± 77.59	45.07 ± 73.01	45.07 ± 73.01	75.07 ± 65.85	73.43 ± 68.31	132.51 ± 76.43
Bank	96.18 ± 0.62	111.16 ± 4.14	97.70 ± 3.91	116.66 ± 5.44	101.26 ± 3.03	100.51 ± 2.80	99.77 ± 5.34	92.83 ± 4.15	94.47 ± 6.13	105.87 ± 2.83
Blood	97.90 ± 1.24	100.92 ± 5.35	93.54 ± 3.96	124.81 ± 9.96	108.82 ± 5.05	107.53 ± 4.02	108.89 ± 3.54	92.25 ± 2.87	92.58 ± 3.13	114.18 ± 4.07
Breast-cancer-wisc-diag	58.60 ± 4.87	49.18 ± 21.88	48.81 ± 21.01	40.40 ± 10.17	27.81 ± 17.56	25.83 ± 16.18	24.22 ± 13.55	35.55 ± 11.22	34.82 ± 15.56	67.21 ± 12.49
Breast-cancer-wisc-prog	98.67 ± 2.27	122.29 ± 24.41	111.32 ± 19.13	120.18 ± 22.65	105.85 ± 15.90	107.68 ± 24.75	108.63 ± 25.97	109.19 ± 20.51	111.40 ± 23.50	127.77 ± 17.95
Breast-cancer-wisc	42.39 ± 7.53	41.10 ± 9.69	49.02 ± 14.54	40.44 ± 18.99	31.30 ± 17.84	32.10 ± 18.23	35.28 ± 15.28	39.94 ± 20.14	38.95 ± 15.35	54.13 ± 18.22
Breast-cancer	94.83 ± 2.43	99.22 ± 10.73	101.11 ± 7.07	117.59 ± 16.06	111.32 ± 11.38	112.77 ± 11.87	115.11 ± 10.81	104.46 ± 11.90	110.64 ± 7.47	118.69 ± 9.24
Breast-tissue	76.84 ± 8.01	77.80 ± 20.83	79.52 ± 23.21	75.89 ± 27.92	98.27 ± 20.29	84.92 ± 24.48	78.96 ± 33.89	75.31 ± 11.32	77.58 ± 13.99	104.08 ± 11.32
Bupa	96.09 ± 2.82	103.16 ± 7.94	109.46 ± 13.07	124.65 ± 8.26	129.90 ± 3.37	129.43 ± 3.63	129.43 ± 3.63	94.47 ± 8.36	94.72 ± 8.45	136.60 ± 11.34
Caesarian	93.60 ± 11.35	96.05 ± 22.13	99.08 ± 10.67	116.36 ± 24.92	135.09 ± 8.43	122.67 ± 20.60	127.78 ± 26.66	122.23 ± 31.46	121.16 ± 27.44	142.18 ± 31.33
Car	95.36 ± 0.79	69.09 ± 2.61	29.29 ± 7.47	36.24 ± 6.01	32.85 ± 8.17	27.62 ± 8.94	26.08 ± 7.74	43.50 ± 5.29	44.30 ± 5.80	114.50 ± 0.15
Cardiotocography-10classes	84.03 ± 0.62	75.38 ± 3.86	60.30 ± 4.24	70.89 ± 4.93	66.78 ± 3.77	64.57 ± 4.93	63.32 ± 3.90	58.39 ± 5.07	60.51 ± 4.16	111.71 ± 2.42
Cardiotocography-3classes	89.05 ± 1.39	94.05 ± 9.53	59.47 ± 10.72	65.38 ± 7.50	69.32 ± 7.48	65.19 ± 8.80	64.76 ± 7.36	59.43 ± 4.46	58.51 ± 8.59	100.16 ± 8.65
Cervical-cancer	84.43 ± 6.46	51.34 ± 37.84	67.07 ± 48.21	57.33 ± 48.43	118.53 ± 2.39	111.60 ± 14.93	111.60 ± 14.93	36.13 ± 34.48	36.27 ± 35.10	101.32 ± 19.29
Chemicalcomposionofceramic	50.06 ± 4.87	0.00 ± 0.00	7.07 ± 22.36	2.46 ± 0.01	147.37 ± 3.13	147.37 ± 3.13	147.37 ± 3.13	1.52 ± 1.09	1.15 ± 0.90	0.00 ± 0.00
Chess-krvk	96.95 ± 0.21	95.51 ± 0.62	92.34 ± 3.10	105.32 ± 3.35	109.87 ± 2.65	107.60 ± 2.33	105.98 ± 3.07	88.64 ± 1.29	90.21 ± 1.38	128.99 ± 1.82
Chess-krvkp	95.07 ± 0.24	63.94 ± 3.46	14.61 ± 7.45	46.19 ± 4.46	21.81 ± 7.50	16.25 ± 8.28	17.13 ± 6.00	14.65 ± 6.53	14.17 ± 7.25	115.92 ± 2.50
Congressional-voting	99.08 ± 0.56	98.72 ± 8.37	100.02 ± 4.89	102.46 ± 10.65	127.09 ± 6.41	127.74 ± 8.38	128.44 ± 9.28	100.83 ± 7.17	100.21 ± 9.10	124.89 ± 4.38
Conn-bench-sonar-mines-rocks	91.67 ± 2.69	105.49 ± 14.07	103.66 ± 12.82	70.39 ± 20.51	77.67 ± 18.97	66.81 ± 28.73	62.39 ± 28.49	76.31 ± 27.58	72.57 ± 27.11	119.98 ± 18.01
Conn-bench-vowel-deterding	75.28 ± 0.72	71.22 ± 4.93	64.24 ± 9.74	8.49 ± 10.29	35.40 ± 6.57	20.36 ± 11.86	13.93 ± 12.42	52.47 ± 6.14	57.37 ± 6.61	119.12 ± 4.56
Connect-4	99.57 ± 0.06	109.71 ± 2.95	94.21 ± 4.89	122.65 ± 5.94	116.17 ± 0.57	112.39 ± 2.25	109.60 ± 4.18	108.75 ± 5.03	106.76 ± 4.24	115.97 ± 0.29

Table 11: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Connectionist	91.28 ± 2.65	106.31 ± 19.31	98.82 ± 14.67	69.96 ± 16.79	123.85 ± 13.85	107.55 ± 25.61	98.53 ± 37.54	72.27 ± 22.00	72.77 ± 21.46	121.69 ± 11.57
Contrac	97.07 ± 0.63	98.84 ± 3.76	101.01 ± 3.41	130.99 ± 4.40	118.62 ± 2.87	118.74 ± 4.60	119.53 ± 4.97	93.08 ± 2.38	93.39 ± 2.68	126.84 ± 2.43
Covid-19	87.72 ± 14.61	79.48 ± 43.86	73.28 ± 73.08	87.99 ± 54.30	82.96 ± 72.68	91.62 ± 83.90	75.86 ± 84.92	70.16 ± 68.24	69.60 ± 71.22	40.14 ± 65.67
Credit-approval	85.23 ± 2.41	87.36 ± 7.07	71.92 ± 8.63	83.63 ± 15.02	75.69 ± 14.75	76.80 ± 12.65	77.99 ± 12.32	71.42 ± 7.89	74.32 ± 8.93	75.63 ± 12.80
Crowdsource	78.25 ± 1.13	78.56 ± 5.38	65.54 ± 4.88	43.75 ± 4.47	100.86 ± 2.14	100.86 ± 2.14	100.86 ± 2.14	51.09 ± 6.51	51.84 ± 7.83	94.05 ± 4.77
Crx	84.30 ± 2.50	87.46 ± 6.08	66.22 ± 8.16	86.43 ± 12.22	133.68 ± 5.46	132.53 ± 6.48	130.46 ± 7.92	77.27 ± 10.24	74.64 ± 6.95	75.63 ± 12.80
Cryother	69.03 ± 11.06	61.61 ± 31.47	40.21 ± 30.25	52.56 ± 36.30	77.05 ± 35.77	74.27 ± 35.31	74.27 ± 35.31	55.57 ± 26.27	64.64 ± 24.75	84.20 ± 23.42
Cylinder-bands	94.25 ± 1.70	109.94 ± 11.74	102.49 ± 12.31	112.91 ± 13.07	100.60 ± 7.85	93.26 ± 9.81	88.51 ± 7.15	98.44 ± 10.03	97.40 ± 9.04	118.22 ± 7.65
Dbworld-bodies	97.96 ± 0.74	91.88 ± 39.78	72.99 ± 50.38	124.94 ± 15.83	134.97 ± 7.40	134.97 ± 7.40	118.84 ± 21.59	/	/	56.43 ± 50.38
Dbworld-bodies-stemmed	97.80 ± 0.83	90.04 ± 36.82	55.89 ± 44.18	115.89 ± 24.62	134.97 ± 7.40	127.54 ± 14.01	100.12 ± 24.08	/	/	88.13 ± 53.62
Dbworld-subjects	98.03 ± 0.74	55.45 ± 29.67	89.11 ± 19.44	84.40 ± 30.41	134.97 ± 7.40	134.97 ± 7.40	134.97 ± 7.40	49.62 ± 40.30	50.30 ± 45.60	120.97 ± 20.46
Dbworld-subjects-stemmed	98.04 ± 0.93	54.72 ± 31.84	78.89 ± 33.77	74.88 ± 28.64	134.97 ± 7.40	134.97 ± 7.40	134.97 ± 7.40	52.27 ± 39.33	54.64 ± 39.86	124.42 ± 15.79
Dermatology	75.32 ± 0.66	18.33 ± 14.16	25.26 ± 16.27	28.11 ± 19.83	18.90 ± 16.82	16.26 ± 17.56	16.01 ± 17.82	19.29 ± 8.81	19.56 ± 9.41	112.15 ± 2.54
Diabetes	90.33 ± 4.04	87.05 ± 8.50	93.05 ± 10.72	114.11 ± 9.22	123.93 ± 0.35	123.93 ± 0.35	123.93 ± 0.35	88.21 ± 5.91	88.85 ± 7.35	111.61 ± 10.35
Diabetic	94.35 ± 1.52	130.35 ± 3.28	102.47 ± 6.98	124.28 ± 7.54	129.33 ± 7.61	127.28 ± 9.96	127.18 ± 9.43	86.96 ± 7.13	87.10 ± 7.01	136.76 ± 8.48
Divorce	30.00 ± 11.71	19.40 ± 25.05	34.83 ± 28.26	17.58 ± 22.38	19.40 ± 25.05	19.40 ± 25.05	19.40 ± 25.05	16.16 ± 20.73	16.27 ± 21.08	32.64 ± 30.09
Dota2train	99.85 ± 0.08	104.85 ± 3.87	119.95 ± 7.33	123.48 ± 5.37	137.75 ± 0.47	136.25 ± 5.32	137.84 ± 4.80	130.90 ± 14.23	128.88 ± 16.59	133.68 ± 5.46
Dow-jones-index	93.85 ± 2.66	110.18 ± 6.50	92.93 ± 6.89	134.37 ± 9.24	138.58 ± 0.30	138.58 ± 0.30	138.58 ± 0.30	105.45 ± 6.32	107.80 ± 5.42	131.99 ± 4.61
Dry-bean-dataset	40.16 ± 1.36	46.24 ± 3.30	39.11 ± 2.97	36.00 ± 3.59	96.74 ± 3.46	96.74 ± 3.46	96.74 ± 3.46	37.71 ± 3.56	38.09 ± 3.06	87.48 ± 2.65
Early-stage-diabetes-data-upload	80.52 ± 2.36	64.82 ± 9.54	38.85 ± 12.41	21.90 ± 19.27	44.74 ± 22.01	42.04 ± 21.76	37.91 ± 23.87	30.79 ± 14.39	32.21 ± 16.84	85.87 ± 10.59
Echocardiogram	91.63 ± 5.20	85.00 ± 32.69	89.39 ± 24.07	105.49 ± 24.24	85.62 ± 32.14	86.34 ± 36.76	86.34 ± 36.76	89.68 ± 23.05	88.82 ± 26.16	78.84 ± 17.81
Ecoli	79.44 ± 1.57	53.34 ± 10.91	59.99 ± 16.71	71.62 ± 12.14	59.84 ± 10.81	59.94 ± 10.23	58.76 ± 8.76	56.03 ± 7.79	58.51 ± 7.77	94.81 ± 10.09
Eegeyesate	92.77 ± 0.51	140.59 ± 12.34	75.92 ± 2.30	81.12 ± 5.42	134.69 ± 0.04	134.63 ± 0.10	134.63 ± 0.10	100.04 ± 2.91	100.40 ± 3.71	122.94 ± 1.94
Electrical	74.59 ± 7.33	28.67 ± 13.76	0.00 ± 0.00	57.61 ± 24.10	86.54 ± 18.21	86.90 ± 16.65	84.50 ± 15.91	17.48 ± 14.95	16.37 ± 13.97	3.84 ± 12.15
Energy-y1	67.33 ± 1.11	66.60 ± 4.27	25.99 ± 7.16	87.63 ± 7.32	60.16 ± 3.73	56.88 ± 4.79	56.45 ± 8.82	53.37 ± 3.02	53.97 ± 4.10	70.36 ± 4.26
Energy-y2	62.98 ± 1.82	59.28 ± 5.49	49.15 ± 6.73	86.64 ± 6.10	57.05 ± 5.91	54.18 ± 4.53	52.63 ± 4.43	46.47 ± 5.18	45.91 ± 6.73	60.36 ± 8.18
Extentionofz-alizadehsani	93.28 ± 1.09	50.69 ± 15.96	4.91 ± 12.18	65.58 ± 17.91	118.38 ± 1.28	118.38 ± 1.28	118.38 ± 1.28	19.52 ± 19.64	19.47 ± 20.01	79.75 ± 16.62
Fertility	99.64 ± 2.50	103.14 ± 8.97	110.92 ± 23.57	116.09 ± 51.45	105.82 ± 2.00	95.33 ± 33.55	89.19 ± 48.84	87.80 ± 34.79	90.72 ± 44.89	105.82 ± 2.00
First-order	97.73 ± 1.16	278.38 ± 9.70	0.00 ± 0.00	21.63 ± 15.46	105.14 ± 0.20	71.37 ± 13.00	45.13 ± 19.14	6.20 ± 9.73	5.63 ± 9.67	103.00 ± 10.21
Flags	93.17 ± 1.73	103.09 ± 6.74	93.28 ± 12.85	114.36 ± 9.25	108.36 ± 10.53	106.34 ± 11.56	107.13 ± 14.89	103.62 ± 13.05	101.17 ± 9.90	103.01 ± 8.41
Foresttypes	65.19 ± 3.23	29.20 ± 21.71	28.54 ± 23.94	29.29 ± 19.57	136.89 ± 2.30	134.86 ± 3.38	134.86 ± 3.38	24.10 ± 16.45	24.09 ± 17.25	67.11 ± 17.23
Garments-worker-productivity	95.85 ± 0.53	105.03 ± 2.13	104.73 ± 4.85	139.38 ± 3.22	130.58 ± 3.01	129.99 ± 3.04	129.23 ± 2.59	99.04 ± 3.33	100.34 ± 3.44	133.66 ± 3.68
Gender-name-dataset	100.32 ± 0.82	124.83 ± 2.80	100.10 ± 0.15	99.11 ± 2.26	122.30 ± 2.35	122.12 ± 3.46	122.02 ± 3.10	100.56 ± 0.65	100.86 ± 0.87	128.32 ± 1.70
Gesture-a1-raw	54.41 ± 2.09	78.70 ± 4.61	48.63 ± 5.56	32.45 ± 5.22	133.07 ± 0.23	133.07 ± 0.23	133.07 ± 0.23	47.38 ± 6.30	47.42 ± 2.85	30.33 ± 6.80
Gesture-a1-va3	83.35 ± 1.72	104.11 ± 4.61	95.81 ± 6.90	87.08 ± 4.64	133.19 ± 0.13	133.19 ± 0.13	133.19 ± 0.13	82.05 ± 2.58	84.43 ± 4.15	106.40 ± 2.71
Gesture-a2-raw	55.12 ± 1.09	86.90 ± 5.39	53.71 ± 7.04	37.23 ± 12.58	132.00 ± 0.25	132.00 ± 0.25	132.00 ± 0.25	49.13 ± 4.49	48.19 ± 6.09	37.91 ± 9.21
Gesture-a2-va3	87.85 ± 2.04	117.67 ± 10.25	105.65 ± 3.66	95.63 ± 6.11	132.16 ± 0.24	132.16 ± 0.24	132.16 ± 0.24	92.42 ± 4.18	94.44 ± 5.19	114.52 ± 5.11
Gesture-a3-raw	47.60 ± 1.73	91.15 ± 3.28	41.91 ± 3.44	31.34 ± 3.97	131.25 ± 0.15	131.25 ± 0.15	131.25 ± 0.15	48.29 ± 2.93	49.41 ± 5.23	35.29 ± 4.54
Gesture-a3-va3	85.41 ± 0.92	111.26 ± 3.87	77.48 ± 6.03	51.91 ± 4.77	131.35 ± 0.20	131.35 ± 0.20	131.35 ± 0.20	83.48 ± 3.53	82.86 ± 2.97	114.95 ± 2.80
Gesture-b1-raw	51.48 ± 2.62	108.43 ± 4.39	43.24 ± 7.92	36.13 ± 10.17	129.61 ± 0.72	129.41 ± 0.81	129.41 ± 0.81	51.59 ± 6.43	48.95 ± 4.28	48.07 ± 6.62
Gesture-b1-va3	87.25 ± 2.47	128.32 ± 7.15	87.85 ± 5.85	50.33 ± 7.39	129.95 ± 0.31	129.95 ± 0.31	129.95 ± 0.31	87.35 ± 3.96	86.40 ± 6.04	125.54 ± 4.73
Gesture-b3-raw	57.69 ± 2.80	93.30 ± 4.39	46.77 ± 8.60	34.08 ± 3.49	132.38 ± 0.32	132.38 ± 0.32	132.38 ± 0.32	50.93 ± 9.12	52.43 ± 8.73	33.85 ± 8.71
Gesture-b3-va3	93.98 ± 1.41	115.66 ± 3.94	101.47 ± 4.61	75.58 ± 4.98	132.48 ± 0.15	132.48 ± 0.15	132.48 ± 0.15	89.32 ± 4.80	94.06 ± 5.10	125.20 ± 5.78
Gesture-c1-raw	57.89 ± 2.51	86.46 ± 3.70	50.93 ± 5.17	37.44 ± 5.75	137.26 ± 0.36	137.26 ± 0.36	137.26 ± 0.36	50.18 ± 7.09	51.27 ± 8.28	38.92 ± 8.27
Gesture-c1-va3	88.63 ± 2.11	102.77 ± 4.22	98.51 ± 4.20	86.33 ± 5.11	137.48 ± 0.28	137.48 ± 0.28	137.48 ± 0.28	85.44 ± 4.29	89.37 ± 2.79	121.76 ± 4.82
Gesture-c3-raw	56.15 ± 2.12	90.95 ± 5.48	53.79 ± 8.13	41.72 ± 7.02	136.91 ± 0.21	136.91 ± 0.21	136.91 ± 0.21	54.09 ± 5.48	59.42 ± 3.46	37.86 ± 4.86

Table 11: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Gesture-c3-va3	91.31 ± 1.49	110.23 ± 2.36	102.38 ± 4.96	97.90 ± 6.11	136.95 ± 0.18	136.95 ± 0.18	136.95 ± 0.18	91.04 ± 3.87	92.53 ± 5.78	121.37 ± 2.94
Glass	82.19 ± 3.64	102.03 ± 7.89	87.40 ± 12.52	87.15 ± 14.20	85.22 ± 10.46	86.50 ± 12.86	88.00 ± 12.60	80.26 ± 10.18	79.87 ± 6.72	107.44 ± 10.85
Go-track-tracks	73.67 ± 8.97	76.28 ± 12.00	60.07 ± 27.30	48.88 ± 36.66	103.74 ± 22.81	104.08 ± 21.33	104.08 ± 21.33	64.23 ± 18.60	64.32 ± 17.64	95.42 ± 28.22
Haberman-survival	97.97 ± 2.89	98.54 ± 7.58	98.53 ± 3.60	128.48 ± 14.74	118.32 ± 10.07	121.39 ± 5.88	121.39 ± 5.88	97.51 ± 6.19	97.41 ± 6.44	117.85 ± 5.87
Hayes-roth	73.71 ± 3.96	77.38 ± 4.16	59.28 ± 10.77	46.15 ± 10.19	60.55 ± 24.53	75.26 ± 13.64	76.81 ± 13.70	73.09 ± 20.45	72.97 ± 21.55	131.15 ± 6.96
Hcc-data	95.40 ± 2.24	109.37 ± 15.30	110.61 ± 12.20	122.62 ± 13.45	127.07 ± 2.32	127.07 ± 2.32	127.07 ± 2.32	111.77 ± 22.45	114.42 ± 17.38	111.74 ± 24.68
Hcvdat	85.09 ± 2.36	78.59 ± 16.68	76.01 ± 11.37	87.36 ± 7.19	104.58 ± 0.23	104.58 ± 0.23	104.58 ± 0.23	64.82 ± 15.36	71.08 ± 13.04	96.31 ± 6.54
Heart-cleveland	92.89 ± 2.29	94.58 ± 10.47	110.50 ± 10.30	117.14 ± 8.26	114.18 ± 10.92	112.30 ± 11.70	114.03 ± 11.93	105.31 ± 8.01	106.11 ± 8.97	120.91 ± 8.17
Heart-hungarian	81.71 ± 5.53	81.94 ± 15.60	85.06 ± 14.95	96.81 ± 17.91	87.03 ± 19.27	85.86 ± 16.91	83.96 ± 18.04	86.49 ± 16.07	87.92 ± 11.64	92.81 ± 23.08
Heart-switzerland	98.89 ± 3.07	106.19 ± 8.26	118.09 ± 14.35	133.59 ± 13.68	130.40 ± 6.78	129.62 ± 6.35	127.68 ± 7.19	111.78 ± 13.06	111.48 ± 11.46	137.20 ± 13.31
Heart-va	98.75 ± 1.41	106.74 ± 8.88	121.36 ± 6.10	128.15 ± 7.90	130.27 ± 9.54	132.80 ± 8.65	134.71 ± 9.31	119.46 ± 7.43	121.70 ± 8.19	137.11 ± 9.38
Heart-failure-clinical-records-dataset	94.23 ± 2.04	87.18 ± 8.48	88.56 ± 11.78	124.71 ± 13.39	121.30 ± 1.49	121.30 ± 1.49	121.30 ± 1.49	102.12 ± 11.40	96.17 ± 18.48	79.54 ± 16.67
Hepatitis	93.31 ± 3.52	91.15 ± 25.26	102.30 ± 29.81	101.25 ± 17.53	99.62 ± 19.31	86.24 ± 47.29	86.24 ± 47.29	95.75 ± 28.21	102.81 ± 31.56	125.06 ± 18.48
Hill-valley	102.00 ± 2.42	139.88 ± 9.03	100.30 ± 0.67	138.51 ± 5.04	138.07 ± 9.33	137.99 ± 10.62	136.58 ± 10.06	96.94 ± 3.62	97.73 ± 3.97	144.77 ± 6.83
Hiv1625data	58.63 ± 3.99	51.93 ± 8.51	64.10 ± 7.23	64.25 ± 5.31	65.88 ± 8.70	60.27 ± 9.60	58.62 ± 10.89	47.93 ± 11.52	57.93 ± 12.01	107.86 ± 4.69
Hiv746data	60.56 ± 5.01	52.90 ± 9.46	73.66 ± 8.01	67.27 ± 8.03	63.74 ± 12.49	59.78 ± 10.06	59.09 ± 11.45	50.21 ± 9.03	51.82 ± 11.75	86.83 ± 11.35
Horse-colic	89.18 ± 2.62	97.44 ± 13.26	70.82 ± 16.01	98.76 ± 16.26	82.97 ± 13.38	80.16 ± 14.45	78.96 ± 16.19	90.78 ± 10.49	87.85 ± 19.79	86.54 ± 12.50
Htru	55.10 ± 3.61	78.10 ± 3.57	47.16 ± 5.91	58.46 ± 3.99	103.17 ± 0.83	101.20 ± 1.13	101.76 ± 1.07	44.44 ± 4.82	44.93 ± 4.79	53.58 ± 4.38
Hypothyroid	93.63 ± 1.15	72.24 ± 6.74	19.79 ± 9.48	107.70 ± 9.64	100.96 ± 1.88	99.96 ± 4.87	100.12 ± 5.11	81.22 ± 19.04	87.35 ± 21.40	71.52 ± 8.75
Ibeacon-rssi-labeled	97.31 ± 0.22	107.16 ± 1.47	94.52 ± 2.03	89.22 ± 2.04	115.45 ± 1.48	113.31 ± 2.53	112.23 ± 2.83	91.45 ± 1.53	91.23 ± 1.74	139.03 ± 1.26
Ilpd-indian-liver	96.22 ± 2.81	144.80 ± 8.99	103.60 ± 6.40	130.56 ± 11.37	118.37 ± 0.65	119.76 ± 3.59	119.36 ± 5.14	93.07 ± 3.12	93.10 ± 3.39	128.80 ± 6.22
Image-segmentation	65.23 ± 2.86	69.31 ± 9.03	45.94 ± 22.03	53.06 ± 10.03	54.39 ± 6.81	53.90 ± 10.25	54.96 ± 10.01	39.77 ± 8.76	42.53 ± 9.91	97.18 ± 11.10
Immunotherapy	93.17 ± 5.79	97.62 ± 21.85	91.37 ± 30.22	124.97 ± 54.46	111.97 ± 4.45	111.97 ± 4.45	111.97 ± 4.45	91.06 ± 29.58	95.88 ± 31.63	89.90 ± 37.42
Impensdata	83.51 ± 2.93	70.69 ± 8.24	100.00 ± 0.00	90.49 ± 9.22	108.93 ± 0.21	108.93 ± 0.21	108.93 ± 0.21	80.25 ± 9.64	85.97 ± 11.92	108.35 ± 6.83
In-vehicle-coupon-recommendation	97.92 ± 0.12	95.44 ± 1.65	92.34 ± 1.19	112.19 ± 2.23	112.59 ± 2.38	111.63 ± 2.38	110.86 ± 2.41	104.27 ± 1.75	103.05 ± 2.88	126.84 ± 1.42
Indian	95.31 ± 2.20	144.46 ± 9.23	105.80 ± 8.41	131.74 ± 12.56	116.21 ± 3.38	117.66 ± 3.01	117.66 ± 3.01	94.53 ± 4.15	94.71 ± 3.40	129.12 ± 6.10
Ionosphere	62.23 ± 4.81	81.13 ± 12.78	59.31 ± 12.13	75.60 ± 13.65	48.69 ± 10.28	43.50 ± 18.93	39.50 ± 17.90	60.43 ± 6.34	59.15 ± 11.10	88.58 ± 22.85
Iris	59.06 ± 8.72	30.32 ± 17.48	22.67 ± 26.19	26.88 ± 26.91	22.36 ± 23.57	24.21 ± 26.11	22.36 ± 23.57	20.87 ± 18.68	21.64 ± 19.95	40.57 ± 28.95
Jain	45.29 ± 13.71	41.53 ± 12.81	5.23 ± 16.55	0.68 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	39.92 ± 11.04	40.90 ± 10.57	44.01 ± 25.18
Jsbach-chorals-harmony	81.57 ± 0.79	50.79 ± 3.84	54.21 ± 4.56	45.20 ± 4.47	60.15 ± 4.88	54.68 ± 4.07	51.89 ± 6.48	45.78 ± 5.80	50.58 ± 8.14	62.55 ± 4.68
Knowledge	85.87 ± 4.18	59.22 ± 16.30	34.90 ± 22.51	81.75 ± 13.88	64.23 ± 21.32	47.97 ± 29.57	38.60 ± 29.57	39.56 ± 17.94	41.39 ± 16.90	67.95 ± 17.68
Lasvegastripadvisorreviews	100.32 ± 0.25	111.50 ± 3.13	121.51 ± 3.80	139.33 ± 3.30	142.23 ± 2.77	143.67 ± 3.93	141.71 ± 4.01	128.63 ± 4.12	129.31 ± 4.45	141.56 ± 2.71
Leaf	86.01 ± 1.17	68.95 ± 10.66	83.50 ± 8.86	135.31 ± 2.22	130.96 ± 3.21	125.23 ± 4.31	124.00 ± 4.63	85.62 ± 6.01	88.69 ± 5.82	125.49 ± 3.94
Led-display	88.07 ± 0.42	66.81 ± 3.97	70.14 ± 4.33	68.69 ± 4.89	78.85 ± 5.79	79.17 ± 5.30	79.60 ± 5.09	68.26 ± 4.39	68.70 ± 4.07	133.70 ± 1.05
Lenses	82.99 ± 20.68	103.60 ± 12.86	60.43 ± 78.78	51.16 ± 53.78	53.59 ± 72.68	55.38 ± 78.02	55.38 ± 78.02	75.56 ± 66.81	79.26 ± 70.36	93.86 ± 84.38
Letter	90.04 ± 0.12	72.31 ± 0.65	47.09 ± 1.30	29.17 ± 1.05	32.64 ± 0.80	28.83 ± 1.07	26.93 ± 0.84	56.10 ± 1.91	58.02 ± 1.50	131.19 ± 0.29
Libras	82.82 ± 2.25	85.98 ± 11.79	74.79 ± 12.39	53.22 ± 10.15	63.27 ± 6.84	55.56 ± 8.53	52.31 ± 8.10	58.94 ± 6.85	58.09 ± 6.86	129.89 ± 5.00
Low-res-spect	79.46 ± 1.62	76.21 ± 7.57	68.59 ± 7.60	69.06 ± 8.72	56.34 ± 6.62	55.72 ± 7.39	53.10 ± 7.30	43.58 ± 7.72	46.68 ± 8.00	88.79 ± 8.96
Lung-cancer	93.99 ± 4.88	96.56 ± 31.00	117.28 ± 29.32	103.21 ± 56.95	110.93 ± 42.92	103.54 ± 41.58	112.79 ± 45.74	114.07 ± 24.85	114.63 ± 25.64	124.19 ± 22.45
Lymphography	87.03 ± 2.34	70.67 ± 17.62	91.72 ± 19.36	88.84 ± 36.07	67.15 ± 29.90	72.22 ± 22.02	77.76 ± 18.30	75.23 ± 17.11	75.45 ± 19.22	94.63 ± 15.89
Magic	84.07 ± 1.78	101.85 ± 4.24	78.74 ± 4.82	93.84 ± 5.47	81.20 ± 5.88	80.57 ± 4.71	80.60 ± 3.86	71.56 ± 5.36	74.11 ± 5.40	110.22 ± 8.88
Mammographic	79.84 ± 2.61	82.69 ± 7.12	72.33 ± 4.82	94.63 ± 9.68	83.20 ± 5.26	84.13 ± 6.19	83.62 ± 6.41	73.77 ± 4.47	75.48 ± 6.23	85.30 ± 2.93
Miniboone	82.25 ± 1.18	189.93 ± 1.93	81.34 ± 7.18	91.38 ± 4.32	88.45 ± 7.42	85.36 ± 6.44	83.07 ± 5.95	75.70 ± 4.34	76.58 ± 4.28	96.24 ± 4.74
Molec-biol-promoter	89.39 ± 4.02	54.41 ± 28.47	96.63 ± 20.34	101.51 ± 23.11	79.01 ± 27.86	71.19 ± 34.52	71.19 ± 34.52	84.66 ± 19.60	85.23 ± 19.65	108.30 ± 20.19

Table 11: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Molec-biol-splice	96.61 ± 0.21	43.89 ± 4.68	47.20 ± 6.16	109.44 ± 3.15	64.55 ± 5.12	64.20 ± 4.41	65.59 ± 3.29	64.73 ± 4.94	65.33 ± 4.39	109.16 ± 3.42
Monks-1	79.68 ± 6.35	85.58 ± 16.00	18.01 ± 25.67	107.12 ± 19.53	77.61 ± 43.67	72.46 ± 41.02	76.16 ± 32.30	43.91 ± 43.45	36.89 ± 47.79	101.45 ± 20.94
Monks-2	100.90 ± 2.75	103.08 ± 4.57	90.54 ± 20.46	126.56 ± 16.70	133.06 ± 10.97	127.05 ± 21.06	121.08 ± 19.59	92.51 ± 20.24	90.88 ± 13.73	132.43 ± 7.25
Monks-3	78.01 ± 7.23	64.86 ± 11.40	40.50 ± 31.32	94.43 ± 30.54	51.83 ± 37.08	57.14 ± 33.62	57.14 ± 33.62	64.71 ± 32.51	55.86 ± 24.15	93.71 ± 8.47
Mushroom	67.75 ± 1.00	65.00 ± 3.45	0.00 ± 0.00	0.03 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.29 ± 0.04	0.22 ± 0.04	23.99 ± 4.20
Musk-1	82.69 ± 2.50	98.13 ± 14.21	75.59 ± 17.22	77.25 ± 12.78	59.52 ± 15.23	54.34 ± 11.78	49.30 ± 12.26	41.77 ± 10.93	42.38 ± 10.08	122.88 ± 14.88
Musk-2	79.29 ± 2.02	102.70 ± 5.93	56.26 ± 8.90	53.27 ± 6.39	53.67 ± 8.49	43.30 ± 8.56	37.44 ± 8.13	26.11 ± 9.61	26.31 ± 10.98	87.81 ± 5.55
Newdiagnosis	50.72 ± 3.08	30.73 ± 7.33	0.00 ± 0.00	0.18 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.51 ± 0.17	1.10 ± 0.10	94.98 ± 16.20
Nursery	89.23 ± 0.16	50.15 ± 1.52	10.57 ± 3.75	47.61 ± 1.53	22.67 ± 1.42	19.67 ± 1.54	17.60 ± 1.60	27.33 ± 4.63	31.85 ± 5.47	92.21 ± 1.63
Obesitydataset-raw-and-data-sinthetic	80.81 ± 0.44	72.54 ± 2.04	36.38 ± 5.59	64.50 ± 4.47	48.72 ± 5.55	43.66 ± 5.84	41.54 ± 6.13	33.04 ± 7.20	34.19 ± 4.07	87.58 ± 4.75
Obs-network-dataset-2-aug27	52.23 ± 2.24	86.99 ± 8.34	2.58 ± 8.15	14.70 ± 8.16	26.00 ± 4.09	26.00 ± 4.09	26.00 ± 4.09	26.01 ± 7.81	21.90 ± 6.73	52.27 ± 6.35
Occupancy-data	35.36 ± 3.13	41.97 ± 5.20	22.80 ± 6.37	18.52 ± 5.69	90.59 ± 4.83	87.73 ± 5.66	87.73 ± 5.66	26.40 ± 5.83	25.76 ± 5.83	22.91 ± 6.78
Occupancy-data2	35.32 ± 2.07	46.60 ± 3.32	17.76 ± 2.81	17.59 ± 3.65	99.54 ± 1.49	92.95 ± 1.88	92.95 ± 1.88	17.73 ± 3.34	17.88 ± 2.64	19.26 ± 2.59
Occupancy-data3	28.48 ± 3.36	33.40 ± 4.77	16.77 ± 1.68	17.60 ± 3.81	101.09 ± 1.58	96.21 ± 1.89	96.17 ± 1.87	21.45 ± 3.16	21.58 ± 2.94	19.14 ± 3.05
Old	85.14 ± 0.57	42.84 ± 3.78	39.55 ± 6.28	26.61 ± 5.34	44.47 ± 4.32	43.75 ± 5.86	42.45 ± 5.52	30.94 ± 5.80	30.90 ± 6.98	67.54 ± 4.81
Online-shoppers-intention	93.64 ± 0.25	107.81 ± 3.48	82.63 ± 2.45	118.80 ± 1.82	108.85 ± 0.19	109.02 ± 0.45	109.22 ± 0.54	78.72 ± 2.32	80.45 ± 4.54	94.78 ± 4.01
Oocytes-merluccius-nucleus-4d	92.39 ± 2.57	126.11 ± 8.34	93.80 ± 4.53	111.77 ± 7.42	101.17 ± 6.69	94.63 ± 5.96	92.41 ± 7.33	80.60 ± 6.86	83.56 ± 8.49	122.17 ± 9.10
Oocytes-merluccius-states-2f	63.53 ± 4.90	79.46 ± 9.81	61.47 ± 11.02	61.45 ± 9.52	59.47 ± 9.30	58.27 ± 7.35	55.66 ± 7.29	53.41 ± 5.97	54.99 ± 7.37	88.04 ± 11.26
Oocytes-trisopterus-nucleus-2f	94.00 ± 2.91	130.24 ± 8.25	98.36 ± 10.84	101.83 ± 11.84	87.53 ± 8.52	81.11 ± 8.90	82.39 ± 6.95	72.86 ± 8.59	78.50 ± 8.77	128.61 ± 9.79
Oocytes-trisopterus-states-5b	71.13 ± 6.12	95.75 ± 8.13	64.57 ± 8.79	58.95 ± 12.75	56.09 ± 12.28	54.47 ± 10.42	53.04 ± 9.19	46.27 ± 9.36	45.98 ± 10.04	89.06 ± 6.25
Optdigits	89.77 ± 0.15	42.09 ± 2.70	43.53 ± 3.37	17.13 ± 4.00	77.23 ± 2.78	74.66 ± 3.13	74.66 ± 3.13	18.30 ± 3.43	16.68 ± 3.29	127.34 ± 1.08
Optical	89.27 ± 0.22	41.47 ± 3.05	45.83 ± 2.96	18.20 ± 3.46	17.43 ± 3.32	16.17 ± 4.42	16.45 ± 3.97	17.67 ± 3.15	18.78 ± 3.43	126.75 ± 1.13
Ozone	99.38 ± 0.13	318.38 ± 16.15	119.39 ± 9.97	131.98 ± 8.63	101.46 ± 0.10	101.46 ± 0.10	101.46 ± 0.10	108.58 ± 14.16	103.74 ± 12.33	106.04 ± 6.86
Page-blocks	73.74 ± 2.77	95.47 ± 6.48	54.63 ± 3.52	64.31 ± 5.03	63.14 ± 6.33	60.92 ± 6.85	60.60 ± 6.90	56.36 ± 6.09	57.36 ± 4.93	81.05 ± 3.97
Parkingbirmingham	49.50 ± 2.79	56.99 ± 1.85	0.00 ± 0.00	22.73 ± 2.99	92.47 ± 4.19	89.49 ± 4.52	89.49 ± 4.52	46.52 ± 1.78	43.82 ± 2.02	0.00 ± 0.00
Parkinsons	80.16 ± 7.40	124.32 ± 25.45	98.43 ± 12.59	36.91 ± 24.59	78.74 ± 21.53	68.60 ± 30.39	62.99 ± 28.27	63.95 ± 25.07	60.75 ± 32.61	83.06 ± 25.05
Pasture	85.52 ± 8.61	70.69 ± 49.92	62.84 ± 47.61	70.45 ± 55.58	145.90 ± 3.85	145.90 ± 3.85	145.90 ± 3.85	63.21 ± 50.80	65.25 ± 51.66	91.28 ± 53.06
Pbc	87.75 ± 3.46	88.39 ± 13.99	84.26 ± 14.91	129.68 ± 13.55	127.52 ± 0.80	127.52 ± 0.80	127.52 ± 0.80	101.73 ± 9.69	98.13 ± 6.58	106.95 ± 18.06
Pen	72.38 ± 0.38	53.31 ± 2.25	26.98 ± 1.93	11.80 ± 1.59	138.66 ± 0.69	137.88 ± 0.77	137.88 ± 0.77	32.79 ± 1.90	33.46 ± 1.55	116.48 ± 1.48
Pendigits	71.98 ± 0.40	49.64 ± 2.71	28.47 ± 2.79	11.05 ± 2.40	8.97 ± 2.47	8.09 ± 3.11	8.23 ± 3.21	30.62 ± 1.93	31.40 ± 1.82	115.72 ± 1.90
Pharynx	101.25 ± 2.56	108.18 ± 9.07	100.00 ± 0.04	139.58 ± 22.58	115.08 ± 16.84	119.68 ± 22.09	126.61 ± 20.52	115.62 ± 9.70	119.82 ± 28.27	201.32 ± 11.50
Phishingwebsites	87.95 ± 0.21	46.34 ± 2.22	37.25 ± 2.06	28.94 ± 2.65	47.16 ± 2.25	44.97 ± 2.23	43.78 ± 2.11	32.02 ± 2.06	32.09 ± 1.80	67.07 ± 1.70
Pima	90.47 ± 3.59	87.25 ± 8.42	93.05 ± 10.72	114.11 ± 9.22	103.18 ± 8.64	104.30 ± 8.58	104.66 ± 7.23	88.21 ± 5.91	88.85 ± 7.35	111.61 ± 10.35
Pittsburg-bridges-rel-l	89.50 ± 5.08	89.37 ± 18.30	97.38 ± 18.60	89.14 ± 38.62	102.52 ± 20.85	101.05 ± 20.80	92.90 ± 21.20	99.94 ± 19.10	96.28 ± 20.16	96.57 ± 19.22
Pittsburg-bridges-span	89.67 ± 4.40	84.73 ± 15.52	94.81 ± 16.21	113.97 ± 26.05	104.74 ± 21.91	105.75 ± 26.40	105.75 ± 26.40	91.20 ± 12.11	91.85 ± 17.20	120.35 ± 16.75
Pittsburg-bridges-t-or-d	95.87 ± 7.30	88.10 ± 22.04	106.33 ± 31.57	106.73 ± 46.76	98.57 ± 35.87	88.59 ± 50.29	84.28 ± 46.41	83.77 ± 58.07	89.37 ± 63.44	98.71 ± 59.34
Pittsburg-bridges-type	92.48 ± 4.29	88.86 ± 10.41	91.56 ± 10.94	101.39 ± 15.88	109.49 ± 15.61	98.66 ± 23.87	98.12 ± 19.29	90.60 ± 19.32	95.59 ± 19.39	107.61 ± 7.39
Pittsburg-bridgesmaterial	80.30 ± 5.59	80.47 ± 16.73	68.43 ± 21.17	81.75 ± 38.26	80.98 ± 15.65	83.67 ± 14.88	82.05 ± 34.00	79.77 ± 19.53	86.96 ± 19.61	78.92 ± 11.95
Planning	100.69 ± 2.28	109.01 ± 8.59	100.00 ± 0.02	127.40 ± 17.80	118.23 ± 1.26	120.18 ± 11.79	128.81 ± 13.19	124.95 ± 13.76	133.30 ± 19.79	135.77 ± 15.55
Plant-margin	95.45 ± 0.12	53.64 ± 4.45	97.15 ± 4.59	70.29 ± 3.67	58.80 ± 4.03	55.28 ± 5.16	54.59 ± 5.18	50.54 ± 3.17	51.14 ± 3.91	136.54 ± 1.61
Plant-shape	93.38 ± 0.12	94.55 ± 3.05	98.45 ± 3.28	81.84 ± 3.95	101.54 ± 1.61	93.41 ± 3.39	89.74 ± 3.79	73.32 ± 4.17	74.65 ± 3.18	135.89 ± 1.20
Plant-texture	95.78 ± 0.15	70.64 ± 4.59	93.55 ± 3.26	61.49 ± 4.34	57.93 ± 4.25	53.86 ± 4.30	52.90 ± 4.50	51.11 ± 2.22	52.36 ± 2.09	139.65 ± 0.65
Poker-hand-training-true	100.00 ± 0.01	100.01 ± 0.04	112.92 ± 2.44	136.31 ± 1.12	120.55 ± 1.45	122.21 ± 1.34	123.81 ± 1.58	99.37 ± 0.59	100.33 ± 1.10	132.74 ± 0.01
Post-operative	100.05 ± 3.10	107.98 ± 7.49	99.93 ± 0.31	135.74 ± 14.49	116.16 ± 4.71	120.44 ± 8.71	124.20 ± 12.12	120.66 ± 17.21	128.98 ± 20.86	120.44 ± 8.71
Primary-tumor	95.96 ± 0.67	92.08 ± 8.27	99.11 ± 5.53	114.86 ± 7.80	109.30 ± 11.30	111.22 ± 10.70	112.01 ± 12.02	102.94 ± 7.17	104.08 ± 7.15	127.72 ± 2.50

Table 11: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Qsarbioconcentration	100.25 ± 0.61	102.46 ± 2.87	103.67 ± 8.00	142.45 ± 10.81	115.78 ± 0.99	119.26 ± 3.67	125.03 ± 6.00	108.72 ± 6.30	110.13 ± 5.28	120.21 ± 6.76
Qsarbiodegradation	89.51 ± 1.69	100.09 ± 6.60	80.51 ± 7.36	85.38 ± 5.91	79.18 ± 7.31	78.18 ± 6.39	79.99 ± 7.23	69.56 ± 5.11	70.77 ± 6.78	101.76 ± 8.68
Qualitative-bankruptcy	32.35 ± 3.73	5.51 ± 10.58	17.29 ± 22.81	1.33 ± 2.38	9.79 ± 21.04	9.79 ± 21.04	4.03 ± 12.74	4.33 ± 11.38	4.13 ± 11.31	13.86 ± 22.80
Ringnorm	74.09 ± 2.09	19.74 ± 2.98	56.68 ± 0.70	99.62 ± 3.20	23.46 ± 3.65	24.51 ± 3.38	25.43 ± 3.77	53.40 ± 4.65	53.76 ± 2.92	119.27 ± 1.63
Risk-factors-cervical-cancer	94.94 ± 1.15	126.72 ± 13.05	77.04 ± 17.52	95.64 ± 19.78	103.33 ± 0.35	104.24 ± 5.08	99.45 ± 11.46	80.68 ± 15.46	87.01 ± 18.10	77.93 ± 21.08
Robotnavigation	55.25 ± 1.48	107.44 ± 1.88	10.13 ± 3.43	59.46 ± 3.13	54.79 ± 3.15	51.52 ± 2.86	49.12 ± 2.39	55.42 ± 4.28	55.26 ± 6.04	86.18 ± 3.25
Sapfile	96.14 ± 1.48	100.24 ± 9.39	115.93 ± 15.89	129.08 ± 8.60	125.60 ± 11.63	124.81 ± 8.63	124.58 ± 10.95	116.65 ± 10.66	117.63 ± 13.94	133.77 ± 14.89
Sat	60.61 ± 0.60	69.58 ± 2.17	56.20 ± 2.03	48.23 ± 2.01	135.02 ± 0.49	133.93 ± 0.43	133.93 ± 0.43	46.77 ± 3.44	49.11 ± 2.28	99.66 ± 1.62
Satelite	60.65 ± 0.61	69.61 ± 2.16	55.92 ± 1.59	48.11 ± 2.83	135.03 ± 0.52	133.86 ± 0.65	133.86 ± 0.65	46.50 ± 1.04	48.25 ± 2.28	99.38 ± 2.04
Scadi	87.09 ± 1.43	61.49 ± 23.59	62.84 ± 26.02	66.13 ± 26.97	79.45 ± 15.60	70.12 ± 31.33	61.52 ± 37.62	67.24 ± 19.91	67.29 ± 20.46	104.98 ± 13.00
Schillingdata	84.94 ± 1.12	62.10 ± 2.63	100.00 ± 0.00	91.64 ± 3.69	107.37 ± 0.09	97.69 ± 4.54	85.26 ± 3.41	71.97 ± 9.52	85.21 ± 12.03	107.98 ± 1.36
Seeds	58.92 ± 5.93	44.58 ± 18.36	45.28 ± 20.77	40.64 ± 6.55	38.07 ± 21.30	38.07 ± 21.30	38.07 ± 21.30	30.45 ± 10.31	29.24 ± 9.57	70.34 ± 19.38
Segment	59.50 ± 1.21	65.40 ± 2.80	25.83 ± 4.34	25.64 ± 2.84	89.82 ± 3.71	87.68 ± 3.17	87.74 ± 3.25	27.25 ± 3.48	25.77 ± 2.60	91.78 ± 3.52
Seismic-bumps	99.02 ± 0.35	138.24 ± 9.17	100.18 ± 0.57	131.00 ± 9.20	103.46 ± 0.01	103.46 ± 0.01	103.46 ± 0.01	100.79 ± 3.96	102.92 ± 4.94	106.98 ± 4.05
Semeion	96.86 ± 0.07	53.86 ± 5.51	70.01 ± 3.45	42.40 ± 4.21	31.05 ± 3.77	29.08 ± 2.48	29.31 ± 2.58	35.85 ± 4.06	35.19 ± 4.68	133.68 ± 0.39
Setapprocesst1	95.41 ± 5.81	130.37 ± 39.71	120.89 ± 28.33	123.25 ± 19.31	128.77 ± 13.55	118.98 ± 21.01	122.56 ± 17.65	116.55 ± 45.33	116.16 ± 44.96	98.22 ± 61.05
Setapprocesst10	99.97 ± 4.17	154.43 ± 21.86	131.27 ± 23.89	127.99 ± 32.23	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	133.00 ± 42.66	133.55 ± 36.13	129.59 ± 14.46
Setapprocesst11	99.41 ± 4.33	134.13 ± 42.31	116.67 ± 34.31	132.74 ± 23.21	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	121.08 ± 25.91	114.69 ± 28.33	137.85 ± 16.93
Setapprocesst2	98.71 ± 4.31	114.63 ± 26.50	90.85 ± 49.65	117.85 ± 48.89	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	118.79 ± 31.46	113.52 ± 32.85	77.95 ± 49.39
Setapprocesst3	99.05 ± 4.05	122.25 ± 23.50	126.61 ± 30.35	119.20 ± 27.49	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	117.27 ± 24.34	118.37 ± 16.90	143.18 ± 26.50
Setapprocesst4	100.78 ± 1.89	146.77 ± 47.89	141.95 ± 29.31	133.29 ± 34.37	116.99 ± 7.71	116.99 ± 7.71	116.99 ± 7.71	136.51 ± 38.59	136.70 ± 30.51	113.48 ± 23.96
Setapprocesst5	97.92 ± 5.06	150.96 ± 26.50	134.24 ± 31.44	129.29 ± 27.69	124.00 ± 9.11	125.95 ± 11.41	125.95 ± 11.41	121.69 ± 19.30	120.29 ± 18.53	113.68 ± 55.27
Setapprocesst6	98.36 ± 2.96	114.55 ± 30.62	116.30 ± 25.27	122.44 ± 27.05	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	117.72 ± 37.58	119.04 ± 33.16	130.04 ± 35.03
Setapprocesst7	98.65 ± 4.55	98.83 ± 35.72	118.41 ± 49.74	121.67 ± 25.15	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	97.17 ± 40.62	98.87 ± 40.41	143.69 ± 23.95
Setapprocesst8	97.57 ± 3.64	113.36 ± 28.07	103.27 ± 30.86	120.07 ± 43.69	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	109.43 ± 40.65	107.74 ± 43.23	150.24 ± 22.05
Setapprocesst9	98.89 ± 3.20	117.62 ± 26.75	110.54 ± 31.35	133.62 ± 14.38	122.05 ± 5.22	122.05 ± 5.22	122.05 ± 5.22	97.88 ± 43.95	108.83 ± 43.44	147.47 ± 28.50
Shillbiddingdataset	68.39 ± 0.58	55.61 ± 4.28	19.11 ± 3.70	28.07 ± 5.14	21.45 ± 4.75	20.85 ± 5.67	18.40 ± 5.37	15.59 ± 4.50	15.05 ± 7.07	53.18 ± 5.20
Shuttle-landing-control	105.16 ± 98.60	166.55 ± 227.78	62.34 ± 15.65	84.84 ± 117.72	10.69 ± 33.80	10.69 ± 33.80	10.69 ± 33.80	102.61 ± 199.52	106.67 ± 212.35	90.69 ± 251.48
Somervillehappinesssurvey2015	97.70 ± 6.04	103.40 ± 12.60	96.69 ± 20.43	111.71 ± 21.10	125.77 ± 16.48	128.01 ± 15.36	127.58 ± 18.64	117.91 ± 19.08	118.43 ± 20.09	114.71 ± 26.82
Sonar	91.62 ± 2.77	105.49 ± 14.10	103.66 ± 12.82	70.39 ± 20.51	116.68 ± 9.43	108.18 ± 16.25	101.43 ± 18.65	74.27 ± 29.19	74.33 ± 29.17	121.44 ± 18.90
Soybean	89.33 ± 0.66	36.08 ± 16.54	47.74 ± 8.71	49.93 ± 10.10	44.37 ± 13.94	39.92 ± 11.59	41.48 ± 12.26	39.47 ± 11.30	39.60 ± 9.78	123.24 ± 2.12
Spambase	85.27 ± 0.75	93.02 ± 3.90	52.34 ± 4.65	62.03 ± 3.75	52.49 ± 3.69	52.01 ± 4.34	52.40 ± 3.66	54.04 ± 3.55	59.05 ± 14.74	95.13 ± 4.38
Speaker-accent	91.78 ± 1.83	94.38 ± 11.13	88.13 ± 14.93	72.21 ± 16.93	109.16 ± 4.98	103.82 ± 5.74	103.82 ± 5.74	67.13 ± 9.34	72.38 ± 10.63	119.12 ± 5.82
Spect	95.44 ± 7.39	102.14 ± 40.33	103.21 ± 21.86	133.24 ± 23.52	124.73 ± 22.08	125.35 ± 24.79	128.37 ± 18.79	128.96 ± 32.08	128.60 ± 34.38	110.24 ± 19.55
Spectf	94.23 ± 3.59	91.65 ± 27.14	98.75 ± 23.79	108.64 ± 30.80	113.48 ± 11.61	113.48 ± 11.61	90.20 ± 38.92	96.18 ± 37.11	96.91 ± 27.49	106.06 ± 16.70
Statlog-australian-credit	98.96 ± 0.48	107.03 ± 5.64	108.70 ± 11.95	142.02 ± 5.56	121.92 ± 3.24	127.38 ± 11.14	128.23 ± 10.04	113.34 ± 6.96	115.26 ± 7.19	125.81 ± 11.36
Statlog-german-credit	97.75 ± 0.31	91.18 ± 9.26	101.98 ± 10.77	123.48 ± 7.63	106.29 ± 9.64	103.65 ± 8.60	103.51 ± 10.34	111.43 ± 6.66	108.76 ± 7.25	117.63 ± 4.61
Statlog-heart	82.60 ± 3.30	72.29 ± 14.04	91.43 ± 18.07	97.58 ± 22.29	82.16 ± 12.91	88.01 ± 19.85	88.40 ± 22.00	85.72 ± 11.79	83.24 ± 17.25	107.56 ± 12.07
Statlog-image	57.69 ± 1.16	65.61 ± 2.80	25.80 ± 4.36	25.64 ± 2.84	36.41 ± 2.85	33.47 ± 3.49	32.41 ± 3.37	26.62 ± 3.57	27.11 ± 4.36	92.01 ± 3.11
Statlog-landsat	60.00 ± 1.20	69.60 ± 3.86	56.29 ± 3.34	49.16 ± 4.47	50.91 ± 3.91	49.15 ± 3.50	48.94 ± 4.15	47.79 ± 2.41	48.96 ± 4.41	98.91 ± 2.10
Statlog-shuttle	75.97 ± 0.52	55.93 ± 2.89	3.96 ± 2.30	5.32 ± 2.37	11.79 ± 1.08	10.62 ± 1.27	10.45 ± 1.18	12.91 ± 0.97	12.44 ± 1.44	54.45 ± 1.53
Statlog-vehicle	82.35 ± 1.47	105.79 ± 3.29	77.33 ± 7.49	89.18 ± 7.10	78.56 ± 6.68	72.42 ± 4.86	72.01 ± 4.41	60.55 ± 4.46	63.31 ± 6.52	113.97 ± 6.04
Steel-plates	81.03 ± 1.11	95.55 ± 4.13	73.83 ± 4.89	84.63 ± 4.10	79.20 ± 4.55	76.86 ± 3.76	76.98 ± 4.79	75.69 ± 4.31	76.97 ± 4.56	115.36 ± 4.37
Synthetic-control	71.91 ± 1.54	33.55 ± 10.45	43.08 ± 8.09	26.76 ± 11.35	8.00 ± 10.33	6.00 ± 9.66	6.00 ± 9.66	10.17 ± 6.45	9.86 ± 6.70	101.50 ± 4.44
Teaching	94.76 ± 3.91	96.13 ± 6.24	98.63 ± 12.16	101.23 ± 17.58	113.25 ± 18.91	106.11 ± 17.95	105.23 ± 17.49	94.76 ± 8.62	97.42 ± 6.28	126.23 ± 13.02
Thoraricsurgery	99.45 ± 0.57	109.78 ± 18.45	102.63 ± 4.26	132.67 ± 17.50	108.40 ± 0.00	110.64 ± 3.62	110.64 ± 3.62	116.89 ± 12.79	114.78 ± 17.81	114.29 ± 5.75

Table 11: Continued.

Dataset	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2	OneR
Thyroid	68.47 ± 12.02	26.24 ± 23.77	48.57 ± 29.18	27.35 ± 21.99	101.08 ± 10.29	98.28 ± 9.98	98.28 ± 9.98	29.59 ± 19.61	29.79 ± 20.14	53.58 ± 35.33
Thyroid-train	93.14 ± 1.53	70.78 ± 7.61	14.42 ± 13.66	105.06 ± 8.86	83.03 ± 6.05	77.92 ± 5.45	72.39 ± 6.23	61.02 ± 11.32	67.36 ± 7.95	70.09 ± 8.43
Tic-tac-toe	96.73 ± 0.39	90.24 ± 1.41	47.77 ± 9.33	0.19 ± 0.01	19.64 ± 15.44	12.30 ± 13.80	5.20 ± 11.16	28.86 ± 7.00	26.86 ± 10.01	115.14 ± 4.59
Titanic	88.41 ± 1.91	89.75 ± 4.30	84.88 ± 3.28	84.21 ± 2.86	99.13 ± 5.01	98.09 ± 4.62	98.09 ± 4.62	86.21 ± 3.78	86.83 ± 3.79	101.12 ± 4.52
Trains	84.82 ± 15.40	73.34 ± 94.66	51.94 ± 84.05	76.67 ± 77.46	73.33 ± 94.67	55.00 ± 88.56	55.00 ± 88.56	50.95 ± 67.29	49.70 ± 68.53	183.33 ± 0.00
Transfusion	97.89 ± 1.28	101.29 ± 5.76	93.54 ± 3.96	123.86 ± 10.03	116.61 ± 6.74	121.88 ± 6.71	124.21 ± 8.00	92.25 ± 2.87	92.58 ± 3.13	114.18 ± 4.07
Trial	71.22 ± 2.99	61.56 ± 12.59	0.00 ± 0.00	3.58 ± 10.41	15.05 ± 13.28	11.72 ± 12.36	11.72 ± 12.36	12.98 ± 13.25	10.62 ± 13.25	0.00 ± 0.00
Turkiye-student-evaluation	94.38 ± 0.27	65.14 ± 3.64	1.60 ± 3.37	59.50 ± 4.97	40.85 ± 2.39	37.12 ± 3.05	37.12 ± 3.05	0.40 ± 0.05	0.29 ± 0.04	1.60 ± 3.37
Unbalanced	99.74 ± 0.37	235.69 ± 70.87	99.98 ± 0.04	124.01 ± 60.40	100.62 ± 0.20	108.95 ± 17.51	115.38 ± 19.81	111.48 ± 11.91	113.44 ± 19.54	100.62 ± 0.20
Urbanlandcover	82.34 ± 3.57	66.22 ± 17.19	65.92 ± 14.08	68.91 ± 17.55	136.97 ± 1.66	136.97 ± 1.66	136.97 ± 1.66	60.94 ± 18.91	61.47 ± 19.17	106.26 ± 15.89
Userknowledgemodeling	84.76 ± 4.06	54.15 ± 11.90	36.41 ± 14.46	70.68 ± 20.81	78.40 ± 11.21	63.64 ± 9.44	62.85 ± 8.75	39.38 ± 12.61	36.40 ± 17.04	61.92 ± 18.43
Vehicle	81.91 ± 1.54	106.66 ± 3.74	77.14 ± 7.64	89.18 ± 7.10	136.15 ± 2.49	135.32 ± 3.04	135.32 ± 3.04	*60.31 ± 5.77*	62.30 ± 8.02	113.14 ± 6.02
Vertebral-column-2classes	85.52 ± 8.10	90.62 ± 18.71	78.76 ± 15.04	89.82 ± 12.51	79.79 ± 17.55	82.11 ± 19.52	81.50 ± 18.11	*68.63 ± 12.67*	68.66 ± 13.18	108.06 ± 17.67
Vertebral-column-3classes	78.56 ± 3.93	62.09 ± 8.69	68.73 ± 9.58	81.65 ± 9.57	70.69 ± 13.71	68.08 ± 15.71	69.92 ± 13.91	*57.42 ± 7.72*	57.94 ± 7.70	88.43 ± 12.33
Veteran	*95.55 ± 1.99*	102.19 ± 14.10	99.16 ± 13.68	136.01 ± 16.45	118.84 ± 0.89	118.84 ± 0.89	118.84 ± 0.89	119.03 ± 25.78	112.21 ± 25.61	111.85 ± 15.81
Vowel	84.58 ± 1.12	72.78 ± 4.98	59.07 ± 7.83	*9.27 ± 8.77*	57.14 ± 5.18	43.98 ± 4.85	39.05 ± 5.81	36.65 ± 7.58	33.50 ± 6.25	122.24 ± 3.70
Wall-following	55.39 ± 1.48	107.47 ± 1.49	*9.26 ± 4.25*	59.65 ± 3.10	57.51 ± 3.68	53.11 ± 4.42	50.98 ± 3.82	57.18 ± 3.46	56.64 ± 4.85	85.73 ± 2.46
Waveform-noise	86.71 ± 1.01	71.40 ± 3.04	83.64 ± 2.55	88.91 ± 2.16	*64.39 ± 3.61*	66.17 ± 2.97	67.24 ± 3.23	66.72 ± 3.80	68.45 ± 3.21	117.79 ± 3.50
Waveform	82.03 ± 0.70	69.94 ± 2.73	81.03 ± 2.77	83.15 ± 3.65	*63.76 ± 3.21*	65.52 ± 3.16	65.98 ± 3.25	64.30 ± 3.07	67.02 ± 2.96	119.31 ± 2.07
Wbc	37.11 ± 6.99	29.48 ± 13.02	49.19 ± 5.07	38.62 ± 12.77	28.23 ± 16.15	28.23 ± 16.15	*25.70 ± 18.48*	39.65 ± 7.43	37.50 ± 9.78	62.02 ± 9.71
Wdbc	58.62 ± 4.93	49.24 ± 21.86	48.65 ± 21.03	40.40 ± 10.17	126.24 ± 0.73	126.24 ± 0.73	126.24 ± 0.73	*33.32 ± 11.34*	35.81 ± 14.00	65.33 ± 10.09
Weathernominal	104.96 ± 33.71	91.92 ± 35.51	89.15 ± 79.96	112.45 ± 59.89	69.39 ± 73.26	94.39 ± 88.11	65.79 ± 91.09	*65.58 ± 89.13*	65.63 ± 89.79	125.02 ± 96.40
Weathernumeric	120.03 ± 26.66	99.85 ± 31.34	*56.97 ± 76.30*	59.08 ± 76.89	69.39 ± 73.26	105.66 ± 96.79	105.66 ± 96.79	60.65 ± 85.06	61.50 ± 85.55	125.02 ± 96.40
Website-phishingdata	79.99 ± 1.06	64.11 ± 4.29	*50.91 ± 5.65*	57.76 ± 5.77	71.44 ± 3.16	70.14 ± 3.00	66.14 ± 5.56	58.22 ± 5.13	57.40 ± 6.15	80.60 ± 3.65
Wholesalecustomersdata	63.07 ± 4.98	63.19 ± 10.85	61.88 ± 9.32	72.69 ± 15.03	121.49 ± 0.83	121.49 ± 0.83	121.49 ± 0.83	*54.15 ± 9.90*	54.77 ± 10.03	64.87 ± 13.39
Wifi-localization	59.41 ± 1.29	*19.68 ± 4.03*	26.37 ± 2.75	20.62 ± 5.50	74.62 ± 4.57	72.06 ± 4.85	72.06 ± 4.85	21.39 ± 2.98	21.94 ± 3.18	74.44 ± 8.05
Wilt	98.84 ± 1.18	121.71 ± 6.00	55.88 ± 7.46	100.06 ± 5.77	101.22 ± 1.27	101.22 ± 1.27	101.22 ± 1.27	54.61 ± 11.19	*51.12 ± 9.33*	104.16 ± 1.59
Wine-quality-red	*89.86 ± 1.73*	97.64 ± 3.22	101.93 ± 3.86	104.40 ± 4.03	107.55 ± 3.49	107.00 ± 3.74	106.82 ± 3.78	92.25 ± 3.08	92.41 ± 3.62	118.72 ± 3.95
Wine-quality-white	*92.27 ± 0.45*	103.72 ± 1.36	102.88 ± 3.21	101.09 ± 3.72	112.80 ± 3.01	111.66 ± 3.27	111.33 ± 3.51	93.01 ± 1.49	93.72 ± 1.18	126.72 ± 2.49
Wine	58.31 ± 4.34	19.42 ± 21.36	37.09 ± 22.77	32.53 ± 22.18	*8.20 ± 17.29*	12.45 ± 20.05	12.45 ± 20.05	19.20 ± 17.00	19.17 ± 17.76	82.06 ± 17.14
Yamilnaduelectricty	*97.44 ± 0.59*	100.01 ± 0.01	115.13 ± 0.78	128.98 ± 1.07	140.56 ± 0.41	140.44 ± 0.42	140.53 ± 0.46	100.05 ± 0.02	100.08 ± 0.02	139.95 ± 0.85
Yeast	93.16 ± 0.64	85.84 ± 2.25	96.02 ± 4.56	110.39 ± 2.76	101.06 ± 5.16	100.97 ± 5.39	100.78 ± 5.69	*85.35 ± 2.58*	86.19 ± 2.76	124.22 ± 3.50
Youtobe-kabita-preprocessing	94.81 ± 0.39	106.30 ± 0.92	101.09 ± 1.73	114.12 ± 1.29	120.91 ± 2.29	121.97 ± 2.09	122.63 ± 2.01	*94.26 ± 0.77*	94.43 ± 0.79	125.61 ± 1.86
Youtobe-nisha-preprocessing	94.07 ± 0.26	102.51 ± 1.20	99.08 ± 1.55	112.16 ± 1.84	117.86 ± 2.11	119.07 ± 2.06	119.47 ± 2.18	*93.15 ± 0.91*	93.46 ± 0.93	126.67 ± 1.21
Z-alizadehsani	96.14 ± 1.07	85.75 ± 15.52	95.62 ± 20.20	101.44 ± 15.72	118.38 ± 1.28	118.38 ± 1.28	118.38 ± 1.28	*82.76 ± 19.42*	85.13 ± 15.70	119.29 ± 10.99
Zoo	69.25 ± 4.04	*17.28 ± 26.77*	29.70 ± 31.08	18.10 ± 20.91	24.24 ± 31.96	24.24 ± 31.96	24.24 ± 31.96	21.01 ± 21.73	20.30 ± 22.40	82.11 ± 16.46
Average rrse (rank)	83.25 (5)	85.51 (6)	70.59 (2)	80.47 (4)	90.03 (9)	88.26 (8)	87.19 (7)	69.93 (1)	70.66 (3)	98.86 (10)
Average rrse std (rank)	3.40 (1)	12.11 (5)	12.26 (6)	13.81 (9)	9.34 (2)	10.67 (3)	11.056 (4)	13.39 (8)	13.99 (10)	12.36 (7)

Table 12: Comparison of accuracy.

	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2
NB	1.0								
J48	0.14	*0.01*							
KNN	0.7	0.17	1.0						
SVM1	1.0	1.0	*0.01*	0.15					
SVM2	1.0	1.0	*0.03*	0.35	1.0				
SVM3	1.0	1.0	0.06	0.48	1.0	1.0			
ANN1	0.29	*0.03*	1.0	1.0	*0.03*	0.09	0.14		
ANN2	0.32	*0.04*	1.0	1.0	*0.03*	0.1	0.17	1.0	
OneR	*0.01*	0.15	*0.0*	*0.0*	0.17	0.06	*0.03*	*0.0*	*0.0*

Table 13: Comparison of kappa statistic.

	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2
NB	0.11								
J48	*0.0*	0.06							
KNN	*0.0*	0.59	0.99						
SVM1	1.0	0.12	*0.0*	*0.0*					
SVM2	1.0	0.56	*0.0*	*0.0*	1.0				
SVM3	0.97	0.82	*0.0*	*0.01*	0.97	1.0			
ANN1	*0.0*	0.06	1.0	0.98	*0.0*	*0.0*	*0.0*		
ANN2	*0.0*	0.07	1.0	0.99	*0.0*	*0.0*	*0.0*	1.0	
OneR	1.0	*0.02*	*0.0*	*0.0*	1.0	0.93	0.74	*0.0*	*0.0*

Table 14: Comparison of root mean squared error.

	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2
NB	1.0								
J48	*0.03*	*0.01*							
KNN	1.0	1.0	0.13						
SVM1	0.54	0.79	*0.0*	0.22					
SVM2	0.83	0.96	*0.0*	0.51	1.0				
SVM3	0.95	0.99	*0.0*	0.71	1.0	1.0			
ANN1	*0.02*	*0.01*	1.0	0.1	*0.0*	*0.0*	*0.0*		
ANN2	*0.04*	*0.01*	1.0	0.16	*0.0*	*0.0*	*0.0*	1.0	
OneR	*0.0*	*0.01*	*0.0*	*0.0*	0.59	0.28	0.14	*0.0*	*0.0*

Table 15: Comparison of mean absolute error.

	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2
NB	*0.0*								
J48	*0.0*	*0.04*							
KNN	*0.0*	0.1	1.0						
SVM1	*0.0*	0.65	0.94	0.99					
SVM2	*0.0*	0.45	0.99	1.0	1.0				
SVM3	*0.0*	0.3	1.0	1.0	1.0	1.0			
ANN1	*0.0*	*0.04*	1.0	1.0	0.95	0.99	1.0		
ANN2	*0.0*	*0.03*	1.0	1.0	0.91	0.98	1.0	1.0	
OneR	*0.0*	1.0	0.09	0.2	0.84	0.67	0.5	0.1	0.07

it is not fair to compare ensemble methods with recall network, a simple and basic algorithm with a single classifier. it is not fair to compare ensemble methods with recall network, a simple and basic algorithm with a single classifier.

5.2. RN vs LSTM. In this paper, ANN shows the best performance on the classification task, but long short-term memory (LSTM) is a next-generation recurrent neural network and is better than traditional neural networks, let alone with the RN algorithm. Though the operation

Table 16: Comparison of relative absolute error.

	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2
NB	*0.0*								
J48	*0.0*	*0.0*							
KNN	*0.0*	*0.0*	1.0						
SVM1	*0.0*	*0.0*	0.84	0.94					
SVM2	*0.0*	*0.0*	0.97	0.99	1.0				
SVM3	*0.0*	*0.0*	0.99	1.0	1.0	1.0			
ANN1	*0.0*	*0.0*	1.0	1.0	0.71	0.91	0.97		
ANN2	*0.0*	*0.0*	1.0	1.0	0.63	0.86	0.94	1.0	
OneR	*0.0*	0.93	*0.0*	*0.0*	0.27	0.11	0.06	*0.0*	*0.0*

Table 17: Comparison of root relative squared error.

	RN	NB	J48	KNN	SVM1	SVM2	SVM3	ANN1	ANN2
NB	1.0								
J48	*0.0*	*0.0*							
KNN	1.0	0.84	0.05						
SVM1	0.48	0.92	*0.0*	0.07					
SVM2	0.85	1.0	*0.0*	0.28	1.0				
SVM3	0.96	1.0	*0.0*	0.5	1.0	1.0			
ANN1	*0.0*	*0.0*	1.0	*0.03*	*0.0*	*0.0*	*0.0*		
ANN2	*0.0*	*0.0*	1.0	0.06	*0.0*	*0.0*	*0.0*	1.0	
OneR	*0.0*	*0.0*	*0.0*	*0.0*	0.13	0.03	*0.01*	*0.0*	*0.0*

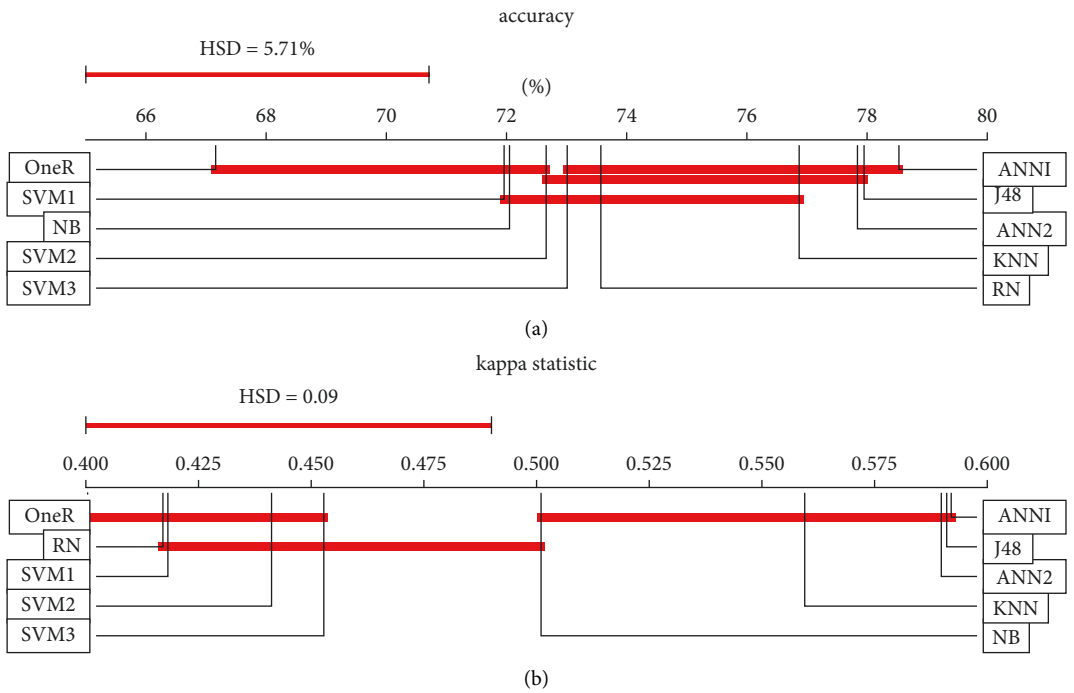


Figure 4: Continued.

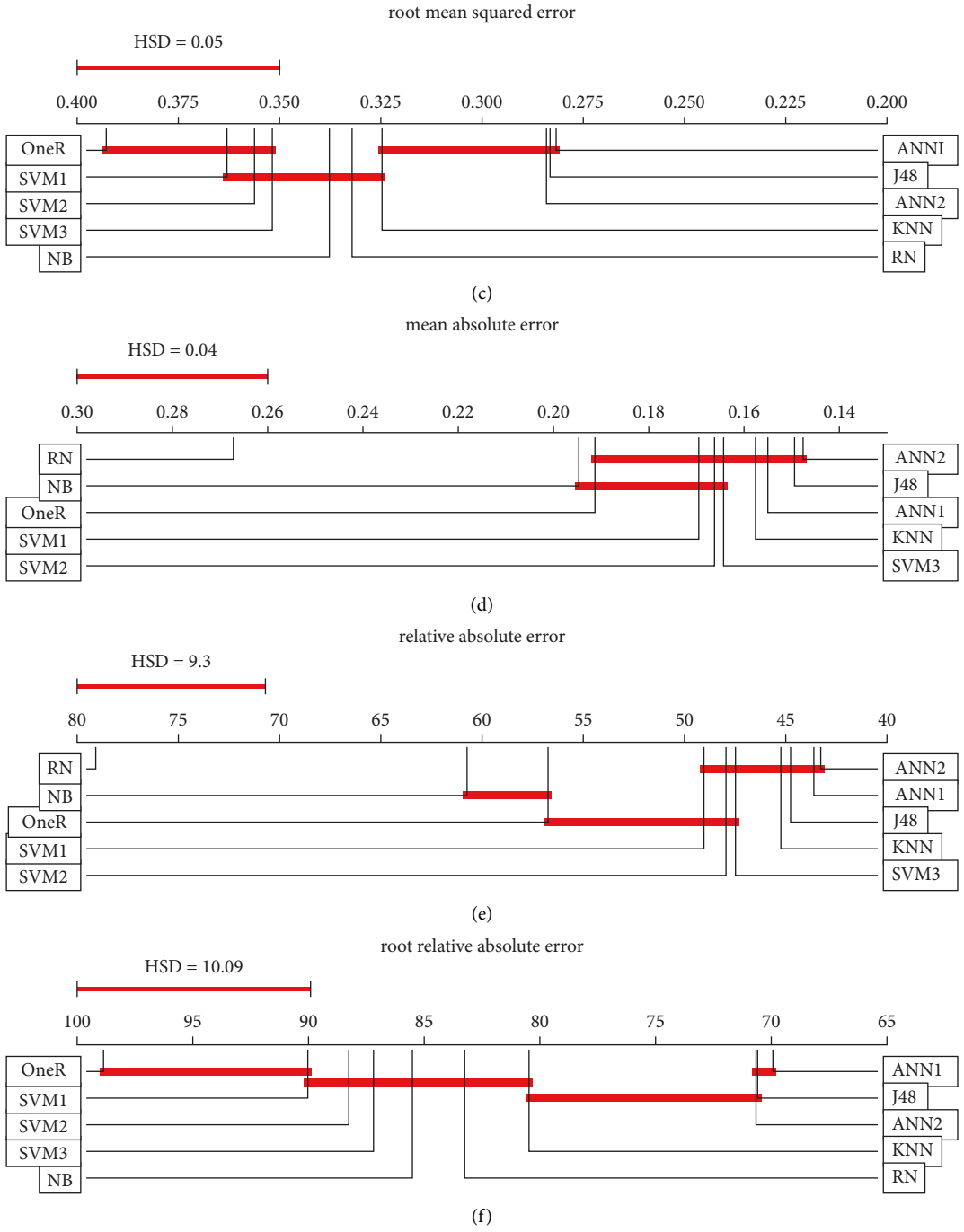


Figure 4: Tukey test results of 6 performance metrics.

mechanism of both LSTM and RN is based on memory, the former obviously employs much more nodes and layers and has more complex structure and therefore takes a much longer running time, so the comparison with LSTM is not to be considered in this work.

5.3. Statistic Tests. In this paper, we employ three types of statistical methods. Friedman test is used to determine distribution differences of the result of the compared algorithms in the six evaluation indicators, while ANOVA is used to determine the mean difference of that. Then, we use the Tukey HSD test to find which pair has significant mean differences.

6. Conclusions

In this paper, a simple and effective classification algorithm is designed as a memory network that stores, marks, and then retrieves the previous paths. Though its structure is similar to artificial neural networks, and the memory mechanism is similar to memory networks, hierarchical temporal memory, and ant colony optimization, the

proposed algorithm is still a naive and distinctive algorithm, and more interpretative than others.

To investigate the capabilities of this newly developed method in the realm of dataset classification, the RN is compared with other classic approaches on benchmark problems in Weka, and experiments show that this simple algorithm has no statistical difference with ANN, J48, KNN, and SVM in accuracy, and all compared algorithms are in the same level on kappa statistic except OneR, though RN performs poorly on RMSE, MAE, RAE, and RRSE.

There are several possible extensions to this work. On the one hand, the classification performance of RN can be improved further, since it is sensitive to the order of layers, so it could be archiving higher performance by exhaustively exploring all possible RN structures and then choosing the best one to fulfill the prediction; moreover, the technique of ensemble learning is also worthy to be used on bagging and boosting recall networks. On the other hand, RN can be also applied in other fields, for example, clustering based on RN could use Hamming distance to partition the routes into small groups.

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 Fundamental Research Funds for National University, China University of Geosciences (Wuhan) (CUG090109).

References

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, pp. 504–507, 2006.

[2] G. E. Hinton, "Learning Multiple Layers of Representation," *Trends in Cognitive sciences*, vol. 11, no. 10, pp. 428–434, 2008.

[3] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Foundations and Trends in Signal Processing*, vol. 7, pp. 197–387, 2014.

[4] J. Weston, S. Chopra, and A. Bordes, "Memory Networks," 2015, https://arxiv.org/abs/1410.3916?context=cs.

[5] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end Memory Networks," 2015, https://arxiv.org/abs/1503.08895?context=cs.

[6] J. Hawkins and S. Blakeslee, *On Intelligence: How a New Understanding of the Brain Will Lead to the Creation of Truly Intelligent Machines*, Henry Holt and Company, New York, NY, U.S.A, 2007.

[7] X. Chen, W. Wang, and W. Li, "An Overview of Hierarchical Temporal Memory: A New Neocortex Algorithm," in *Proceedings of the International Conference on Modelling, Identification and Control*, pp. 1004–1010, Wuhan, China, June 2012.

[8] B. Hu, Z.-H. Guan, G. Chen, and C. L. P. Chen, "Neuroscience and network dynamics toward brain-inspired intelligence," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[9] I. Witten, E. Frank, M. Hall, and C. Pal, *Data Mining: Practical Machine Learning Tools and Techniques, the Morgan Kaufmann Series in Data Management Systems*, Elsevier Science, Netherlands, 2016.

[10] T. Mitchell, *Machine Learning*, pp. 10–16, McGraw-Hill Education, New York, NY, U.S.A, 1997.

[11] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[12] K. He and J. Sun, "Convolutional Neural Networks at Constrained Time Cost," 2014, https://arxiv.org/abs/1412.1710.

[13] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, https://arxiv.org/abs/1502.03167.

[14] C. Szegedy, W. Liu, Y. Jia et al., "Going Deeper with Convolutions," 2014, https://arxiv.org/abs/1409.4842.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the Inception Architecture for Computer Vision*, https://arxiv.org/abs/1512.00567?context=cs, 2015.

[16] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," 2017, https://arxiv.org/abs/1706.03762?context=cs.

[17] J. Weston, S. Chopra, and A. Bordes, "Memory Networks," 2015, https://arxiv.org/abs/1410.3916?context=cs.

[18] R. Arora and S. Suman, "Comparative analysis of classification algorithms on different datasets using weka," *International Journal of Computer Application*, vol. 54, no. 13, pp. 21–25, 2012.

[19] S. A. Kiranmai and A. J. Laxmi, "Data Mining for Classification of Power Quality Problems Using Weka and the Effect of Attributes on Classification Accuracy," *Protection and Control of Modern Power Systems*, vol. 3, no. 1, pp. 1–12, 2018.

[20] V. Mhetre and M. Nagar, "Classification based data mining algorithms to predict slow, average and fast learners in educational system using weka," in *Proceedings of the 2017 International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, July 2018.

[21] S. Farhat, M. Abdelkader, A. M. Makhlouf, and F. Zarai, "Comparative study of classification algorithms for cloud ids using nsl-kdd dataset in weka," in *Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 445–450, IEEE, Limassol, Cyprus, June 2020.

[22] C. N. Villavicencio, J. J. E. Macrohon, X. A. Inbaraj, J. H. Jeng, and J. G. Hsieh, "Covid-19 prediction applying supervised machine learning algorithms with comparative analysis using weka," *Algorithms*, vol. 14, no. 7, p. 201, 2021.

[23] M. Othman and T. Yau, "Comparison of Different Classification Techniques Using Weka for Breast Cancer," in *Proceedings of the 3rd Kuala Lumpur international conference on biomedical engineering 2006*, pp. 520–523, Berlin, Germany, January 2007.

[24] V. Vakharia, I. E. Castelli, K. Bhavsar, and A. Solanki, "Bandgap Prediction of Metal Halide Perovskites Using Regression Machine Learning Models," *Physics Letters A*, vol. 422, 2021.

[25] M. R. Barusu, "Diagnosis of bearing outer race faults using a low-cost non-contact method with advanced wavelet transforms," *Electronics*, vol. 25, no. 1, pp. 44–53, 2018.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1482250, 7 pages
https://doi.org/10.1155/2022/1482250

Research Article

Research on Clustering Algorithm Based on Improved SOM Neural Network

Chengxiang Shi and Xiaoqing Li

Department of Mathematics and Information Engineering, Chongqing University of Education, Chongqing, China

Correspondence should be addressed to Chengxiang Shi; shicx@cque.edu.cn

Received 12 May 2022; Revised 21 June 2022; Accepted 4 July 2022; Published 10 August 2022

Academic Editor: Diego Oliva

Clustering algorithm is a statistical method to study sample classification. With the rapid development of science and technology, people have higher and higher requirements for data classification, so there are more and more researches on clustering in modern society. Various mathematical algorithms are introduced to further improve the accuracy of clustering. Therefore, this paper proposes an improved SOM neural network algorithm to evaluate the comprehensive quality of students. SOM neural network can automatically find the internal laws and essential attributes in the samples, self-organize and adaptively change the network parameters and structure, and realize the classification of samples. Factor analysis is introduced to reduce the dimension of input layer in SOM neural network analysis, better process high-dimensional data, and improve the speed and accuracy of the algorithm. The improved SOM neural network algorithm can be used for the cluster analysis of the comprehensive quality of college students. The algorithm simulation results show that the improved neural network algorithm can intuitively evaluate the comprehensive quality of students and reflect the overall characteristics of each type of student.

1. Introduction

With the advent of the era of big data, the sources of data are becoming richer and richer, and the amount of data also shows a trend of rapid growth. Research and mining of important information contained in data have become a specialty. At present, data mining technology is widely used in various fields, such as economy, finance, transportation, commerce, and education. Cluster analysis is also an important task in data mining. It can find out the laws in the data and express them in the form of visualization. At present, there are many applications of data mining in the field of education, such as students' comprehensive quality evaluation. These assessments are also an important basis for students to strengthen learning, teachers to adjust teaching, and schools to arrange courses.

There are many methods for the evaluation of students' comprehensive quality, such as the analytic hierarchy process adopted by Lin [1], the adaptive multiminimum support association algorithm and SOM neural network algorithm of Xie [2], and the SVM method used by Yang et al. [3].

The SOM neural network adopted in this paper is also widely used in practical life. For example, Chen [4] improved the clustering algorithm SOM-K-means to crawl and classify the network water army, which is of great significance to the governance of the network water army. Wu [5] proposed an improved clustering algorithm, SOM-K-medoids-CH, which can effectively and accurately divide a large number of bank customers, mine out their potential needs, and sell the right products to the right customers at the right time.

However, we find that the data for evaluating students are multidimensional, the subject scores are diverse, and the correlation between subjects is relatively complex [6]. Students can be divided into different categories by directly using the clustering method according to the data [7]. However, for researchers, it is difficult to directly observe the commonalities between each type of student from the classification results because of the large and complex data. Moreover, for SOM neural network algorithm, the result is also greatly affected by the input samples [8]. Therefore, in view of the above problems, this paper will introduce factor analysis into the SOM algorithm model to eliminate the

relevant influence, extract the important indicators in the data, and analyze and verify the classification results.

2. Related Algorithm Theory

2.1. Basic Theory of Factor Analysis. Factor analysis was first proposed by British psychologist C. E. Spearman. In his research, he found that there was a certain correlation between students' grades in various subjects and then speculated whether there were some potential common factors affecting students' academic performance. Factor analysis can find out the hidden representative factors in many variables and classify the variables with the same essence into one factor, which can reduce the number of variables and test the hypothesis of the relationship between variables [9–11].

In factor analysis, each factor is not related to each other, and all variables can be expressed as a linear combination of common factors. There n are samples p and indicators, $X = (X_1, X_2, \ldots, X_P)^T$ which are random vectors. If the common factor to $F = (F_1, F_2, \ldots, F_m)^T$ be found is, the factor model is

$$\begin{aligned} X_1 &= a_{11}F_1 + a_{12}F_2 + \cdots + a_{1m}F_m + \varepsilon_1, \\ X_2 &= a_{21}F_1 + a_{22}F_2 + \cdots + a_{2m}F_m + \varepsilon_2, \\ &\cdots\cdots \\ X_p &= a_{p1}F_1 + a_{p2}F_2 + \cdots + a_{pm}F_m + \varepsilon_p. \end{aligned} \tag{1}$$

The matrix $A = (a_{ij})$ is called the factor a_{ij} load matrix, which reflects the i importance X_i of j the variable F_j to the common factor ε. As a special factor, it represents the variation of variables caused by influencing factors other than common factors, which can be ignored in the practical analysis [12, 13].

The model obtained by factor analysis is not affected by dimension, and its factor load is not unique. When the factor load is complex and difficult to be explained reasonably, a new factor load matrix can be obtained by factor rotation, and its analysis significance will be more obvious.

2.2. Self-Organizing Mapping Network. Self-organizing feature mapping network was proposed by Professor T. Kohonen of Helsinki University in Finland in 1981, which is called SOM network for short. Kohonen believes that when a neural network receives external input, each region of the neural network will have different response characteristics, and this process is completed automatically.

A typical feature of a feature mapping network is that it can be divided into input layer and competition layer on a one-dimensional or two-dimensional processing unit array. After self-organizing training, neurons will be orderly arranged in the competition layer. Neurons with similar functions are very close, and neurons with different functions are far away.

SOM network adopts the Kohonen algorithm, and the influence of winning neurons on their adjacent neurons is from near to far, from excitation to inhibition. Therefore, not only the winning neurons need to adjust the weight but also the surrounding neurons will adjust the corresponding weight. The learning algorithm steps are as follows:

(1) Network initialization, set the initial value of the weight between the input layer and the mapping layer with a random number.

(2) Normalized data and input data. Normalize the data and input the $x = (x_1, x_2, x_3, \ldots x_n)^T$ vector to the input layer.

(3) Calculate the distance between the weight vector of the mapping layer and the input vector. The distance between j the second neuron of the mapping layer and the inputvector is

$$\begin{aligned} d_j &= \left\| X - W_j \right\| \\ &= \sqrt{\sum_{i=1}^{m} \left(x_i(t) - w_{ij}(t) \right)^2}. \end{aligned} \tag{2}$$

where is the weight between i the neurons of the input layer j and the neurons of the mapping layer.

(4) Define areas of excellence.

(5) Weight learning. The weights of winning neurons and adjacent neurons are updated according to the following formula:

$$\Delta w_{ij} = \eta h\left(j, j^*\right)\left(x_i - w_{ij}\right), \tag{3}$$

where η is a constant of

$$h\left(j, j^*\right) = \exp\left(-\frac{\left| j - j^* \right|^2}{\sigma^2} \right), \tag{4}$$

σ^2 decreases with the progress of this learning.

(6) Calculate the $o_k = f\left(\min_j \| X - W_{ij} \|\right)$ output.

(7) If the requirements are met, output the results, otherwise return to (3) to continue.

3. Improved SOM Learning Algorithm

In the improved SOM algorithm, a factor analysis layer is added before the input of SOM sample data. After data are input into factor analysis layer, the factor load matrix table can be obtained by dimensionality reduction of data through factor analysis. By observing the load matrix table, we can get the commonness of each factor after dimensionality reduction and then extract the representative factor and name the representative factor according to the commonness. Then, the extracted data are input into the input layer of the SOM model, and the data are transmitted to the neurons of each competing layer [14, 15]. The improved SOM neural network model is shown in Figure 1.

The first layer is factor analysis. By n inputting samples p and indicators, $X = (X_1, X_2, \ldots, X_P)^T$ the dimensionality of the data is reduced and standardized, the $F = (F_1, F_2, \ldots, F_m)^T$ factors are output, and the factors are named.

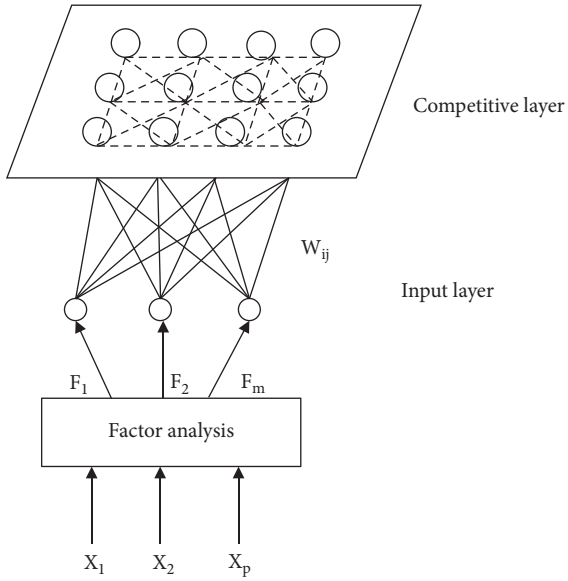


Figure 1: Improved SOM neural network model.

The second layer is the input layer, which is equivalent to a transfer station. It connects the processed data with the competitive layer and is responsible for transmission.

The third layer is the competition layer. The normalized data find the winning neuron by calculating the distance between the weight vector and the input vector of the mapping layer, update the weight of the adjacent neuron, and output the result after judging that it meets the conditions.

4. Empirical Analysis

The data in this paper come from the academic administration system of a certain college in a certain university to obtain the four-year academic performance information tables of 130 students of a certain major in 2016.

4.1. Factor Analysis Data Processing. First, the data of students' specific course records in the grade information table are cleaned. After data processing, the practical courses are combined into practical courses, and the common professional basic courses, professional core courses, and public compulsory courses are selected. Second, eliminate elective courses, screen and modify course name errors, remove missing exams, registration errors, and other noise data, and supplement a few missing grades with 60 points. The final data include variables such as student number, course name, and course score, and 37-course scores are obtained. According to the factor analysis theory, the experiment has 130 samples and 37 indicators, which $X = (X_1, X_2, \ldots, X_{37})^{130}$ are random vectors, and the common factor to be sought is $F = (F_1, F_2, \ldots, F_m)^{130}$.

This section adopts the factor analysis method, and the software used is SPSS statistics 26.

First, the data are imported into the software for factor analysis. After standardizing the data, the KMO value is 0.879, greater than 0.5, and the significance level is significantly less than 0.05, indicating that the variables in this study are suitable for factor analysis. The output results are shown in Table 1.

Then, factor analysis was carried out on all variables to obtain the eigenvalues, variance contribution rate, and cumulative variance contribution rate of 37 variables. According to the research, the components with eigenvalues greater than 1 are selected as factors, and a total of 9 factors are extracted. As shown in Table 2, the cumulative contribution rate of the nine factors is 67.45%, more than 60%, which meets the requirements of factor analysis. The study can extract these nine factors.

The evaluation is based on the notice of the measures for the evaluation of students' comprehensive quality issued by a school, which is also the principle that this study should follow.

From the study of the component matrix of factor analysis, it is found that the common factors displayed by the component matrix are not obvious, and the interpretation of the common factors is slightly difficult. Therefore, in this study, the maximum variance method is used to rotate the component matrix and sort it by size to obtain the rotated component matrix.

Through the total variance interpretation after rotation, 9 factors are obtained, respectively, $F_1, F_2, \ldots, F_8, F_9$. The factors are then named by the rotated matrix list of components. Sort the variables contained in each factor, find out the variables with larger data in the matrix table, observe the commonness between variables, and then get the name of each factor. The resulting factor naming table is shown in Table 3.

4.2. SOM Neural Network Model Analysis. This paper uses MATLAB software to input the obtained data into the software for operation [16].

It can be seen from the input samples that the number of input neurons is 37. This study uses the hexagonal topology output. In the establishment of output layer neurons, there is no authoritative and effective theoretical method, so the trial-and-error method is used to establish the output layer neurons. Through many attempts, the number of output layer neurons is determined as 4, and the two-dimensional 2×2 SOM competition layer neurons are used as the capacity of clustering. The hexagonal topology is shown in Figure 2.

In the confirmation of training times, we can determine from the stability of the classification of training times. In this paper, the data are trained for 10, 25, 50, 100, 200, 500, and 1000 times, respectively, and the classification results after training are obtained. When the training times are 100 times, the classification results have been stable. Therefore, the training frequency of the study is 100 times. The training classification results are shown in Figure 3.

In other initial parameters, the default value of the topology function is "hextop," and the default value of the distance function is "linkdish." After all structures and initial parameters are established, the data are substituted into

Table 1: KMO and bartlett test.

KMO and bartlett test		
KMO sampling suitability quantity		0.879
Bartlett sphericity test	Approximate chi-square	2650.471
	Freedom	666
	Significance	0.000

Table 2: Interpretation of total variance after rotation.

Component	Initial eigenvalue			Sum of squares of rotating loads		
	Total	Percentage variance	Cumulative (%)	Total	Percentage variance	Cumulative (%)
1	12.371	33.434	33.434	9.281	25.085	25.085
2	2.623	7.089	40.523	2.451	6.625	31.709
3	2.353	6.360	46.883	2.447	6.614	38.323
4	1.571	4.245	51.128	2.269	6.132	44.455
5	1.516	4.096	55.224	1.952	5.276	49.731
6	1.277	3.451	58.675	1.950	5.271	55.002
7	1.190	3.217	61.892	1.780	4.812	59.814
8	1.049	2.836	64.728	1.429	3.861	63.675
9	1.010	2.729	67.457	1.399	3.782	67.457
10	.....	.....	.....	.....	.....	.....

Table 3: Factor naming table.

Component	Name	Factor naming
1	F1	Professional core competence factor
2	F2	Innovation and entrepreneurship ability factor
3	F3	Computer capability factor
4	F4	Physical literacy ability factor 1
5	F5	Logical thinking ability factor
6	F6	Political literacy ability factor
7	F7	Language expression ability factor
8	F8	Physical literacy ability factor 2
9	F9	Mental health ability factor

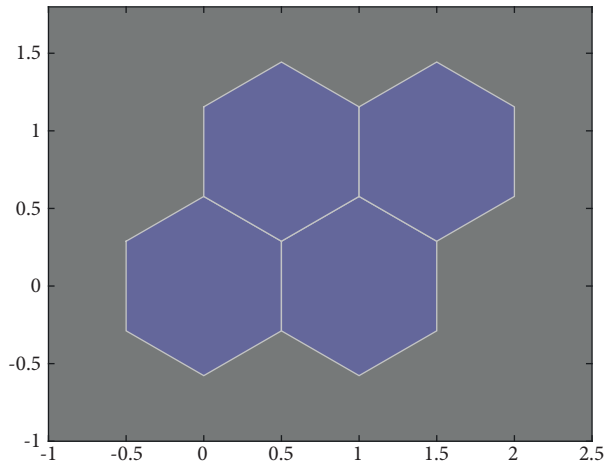


Figure 2: Hexagonal topology.

SOM network training. SOM network automatically looks for the nearest output neuron, finds the winning neuron, and records it. After reaching the training times, SOM clustering training is completed as shown in Table 4.

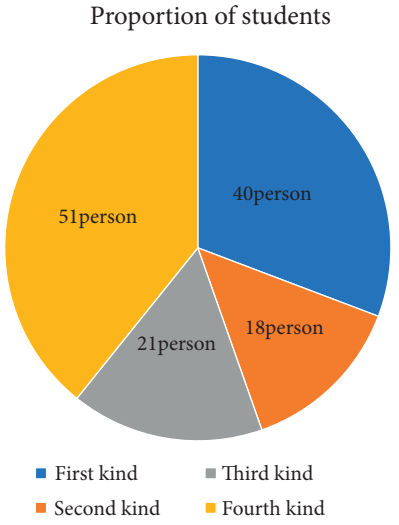


Figure 3: Classification results.

Through SOM neural network analysis, student groups can be divided into four categories. In order to more intuitively observe the proportion of students in each category, a pie chart of the proportion of students is drawn. At this time, we only get the number of people in each category, but the characteristics of these four categories are not known at present, so we will focus on exploring the characteristics of the four groups of people for analysis. The number and proportion of each category are shown in Figure 3.

Through the results of factor analysis in the previous article, the scores of students in each subject and the load after rotation are calculated, and the results are standardized to obtain the nine-dimensional comprehensive quality score of each student. Then, according to the analysis results of the SOM neural network, the students are divided into four categories, and the average value of nine-dimensional

Table 4: Training classification results.

Number of training	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
10	2	4	1	3	1	3	2	3	2	2	1	1	3	1	4	1	3	3	...
25	3	1	4	2	4	2	3	2	3	3	4	4	2	4	1	4	2	3	...
50	3	4	1	2	1	2	3	2	3	3	1	1	2	1	4	1	2	2	...
100	3	4	1	2	1	2	3	2	3	3	1	1	2	1	4	1	2	2	...

Table 5: Average value of comprehensive quality of four types of students.

Name	Classification	One	Two	Three	Four
	Number of students	40	18	21	51
F1	Professional core competence	1.075113	0.1671	0.096766	−0.94205
F2	Innovation and entrepreneurship	0.096096	−0.15272	0.396673	−0.1848
F3	Computer capability	0.216969	−0.10328	0.313487	−0.2628
F4	Physical literacy ability 1	0.11106	−0.65692	0.223025	0.052913
F5	Mathematical logical thinking ability	0.269041	0.342395	−0.87165	0.027059
F6	Political literacy ability	0.237592	−0.20436	0.106884	−0.15823
F7	Language expression ability	0.305684	−0.493	0.104206	−0.10866
F8	Physical literacy ability 2	0.10712	0.222701	−0.29254	−0.04216
F9	Mental health ability	0.125481	0.366019	−0.24545	−0.12653

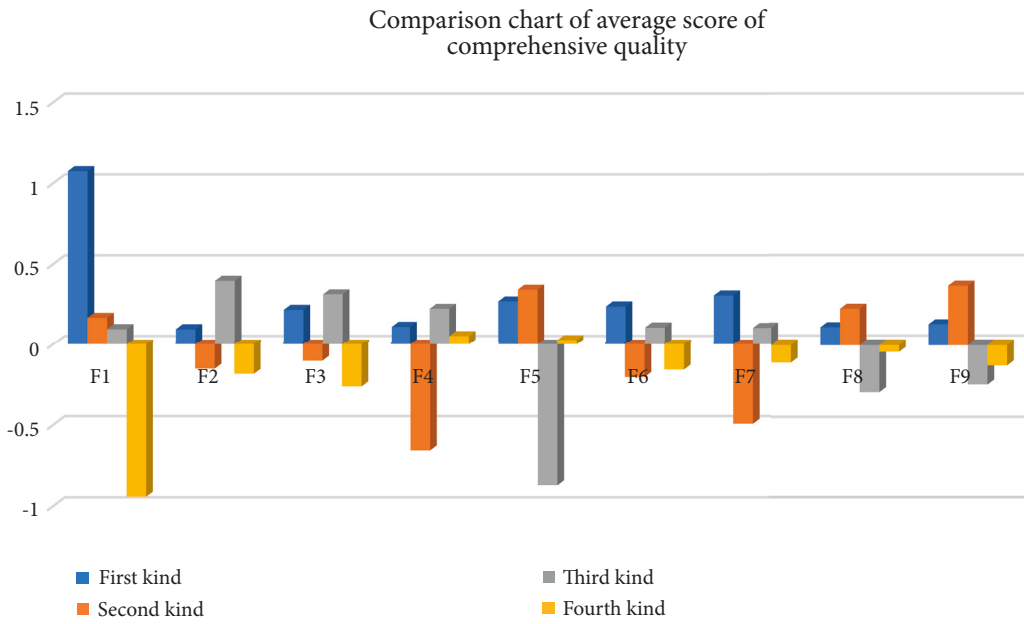


Figure 4: Comparison chart of average score of comprehensive quality.

comprehensive quality indexes of each category of students is calculated [17, 18]. The statistical data obtained are shown in Table 5.

In order to more intuitively observe the characteristics of each type of student, the average value of the comprehensive quality of the four types of students in Table 4 is converted into a bar chart. The abscissa represents each type of comprehensive quality, the ordinate represents the score of comprehensive quality, and different colors represent each type of student group. Figure 4 shows the results.

The data in the table have been standardized, and the average value of each comprehensive quality is 0. Therefore, it can be seen from the above table and figure.

Compared with the top 40 students in this category, all of them have outstanding abilities.

There are 18 students in the second category. These students have obvious deficiencies in innovation and entrepreneurship ability, computer ability, physical quality, and language expression, but their professional core competence is relatively good.

There are 21 students in the third category. Their physical quality and mental health are relatively weak, and their scores in other aspects are higher than those in other categories, except for mathematical logical thinking ability. It can be seen that this kind of student's professional core ability is not strong.

There are 51 students in the fourth category, which is also the largest category. In addition to physical quality and mathematical logical thinking, the rest of these students are relatively low, indicating that they have obvious deficiencies and need to start from the foundation.

5. Conclusion

Through empirical analysis, the algorithm first classifies the students' comprehensive quality into nine categories based on the students' course scores by factor analysis, and the individual students can be evaluated by the classified data. Then, on this basis, SOM neural network clustering analysis is carried out, and students are divided into four categories. Students of different categories have corresponding characteristics, which can be evaluated for different student groups.

Aiming at the limitations of evaluating students' quality, the complexity of various data, and the evaluation based on the total score, this paper puts forward an improved SOM neural network model and adds factor analysis to the model. The model can not only extract the common factors in various disciplines, integrate various comprehensive abilities of students, but also improve the accuracy of clustering. The improved SOM model can evaluate the comprehensive quality of each type of student more intuitively and accurately and provide a strong basis for schools, teachers, and self-management, so as to promote the all-round development of students.

The improved SOM neural network algorithm is of great significance to the evaluation of students' comprehensive quality. The algorithm can reduce dimension and cluster data. However, when there are too many data dimensions, the operation difficulty of this model will increase, which also needs further improvement in the future. The algorithm can be applied in many aspects, not only to analyze students' comprehensive quality but also to evaluate and classify patients in hospitals. It is expected that the algorithm can be improved in the future, so as to make a more perfect evaluation of the comprehensive quality of students and evaluate the development of each student.

Data Availability

All data, models, and code generated or used during the study appear in the submitted article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was partly financially supported through grants from the Chongqing Science and Technology Bureau Technology Innovation and Application Development Key Project (No. cstc2020jscx-dxwtBX0044), the Chongqing Science and Technology Bureau Technology Innovation and Application Development General Project (No. cstc2020jscx-msxmX0152), the Chongqing Special Key Project for Technological Innovation and Application Development (No. cstc2021jscx-dxwtBX0022), the Scientific Research Project of Chongqing University of Education (No. KY202107B), and the University Student Research Project of Chongqing University of Education (No. KY20210166).

References

[1] Y. Lin, *Research and Implementation of College Students' Comprehensive Quality Evaluation System Based on Fuzzy Comprehensive Evaluation*, University of Electronic Science and technology, Sichuan, China, 2008.

[2] Y. Xie, *Research on Curriculum Relevance and Students' Comprehensive Quality Evaluation Based on Students' Achievements*, Central China Normal University, Hubei, China, 2019.

[3] B. Yang, L. Zhang, J. Lin, W. Wang, and P. Xue, "Research on comprehensive quality evaluation method of college students based on SVM," *Computer and information technology*, vol. 28, no. 3, pp. 68–70, 2020.

[4] G. Chen, *Cluster Analysis of Tianya BBS Water Military Posts Based on Som-K-Means*, Huazhong University of science and technology, Huazhong, China, 2013.

[5] H. Wu, *Research on Bank Customer Segmentation Based on Improved SOM*, Changchun University of technology, Changchun, China, 2021.

[6] M. Wang and X. Wu, "Research on the innovation of comprehensive quality evaluation mechanism of college students in the era of big data," *Chinese Journal of multimedia and network teaching (zhongxunjian)*, no. 5, pp. 143–145, 2021.

[7] Y. Zhang, "Research on the comprehensive quality evaluation system of college students in Application-oriented Universities under the background of big data era," *Journal of Shanxi Institute of energy*, vol. 3, no. 1, pp. 34–36, 2021.

[8] I. Y. Purbasari, E. Y. Puspaningrum, and A. Putra, "Using self-organizing map (SOM) for clustering and visualization of new students based on grades," *Journal of Physics: Conference Series*, vol. 1569, no. 2, Article ID 022037, 2020.

[9] Y. Kang and Y. Wang, "Application of principal component analysis in comprehensive evaluation of College Students' physical health," *Journal of Shanxi Normal University (Philosophy and Social Sciences edition)*, vol. 39, no. 3, pp. 30–33, 2019.

[10] X. Xu and L. Chen, "Discussion on student achievement evaluation based on factor analysis and cluster analysis on the cultivation of preventive medicine professionals," *Medical education research and practice*, vol. 29, no. 5, pp. 675–678, 2021.

[11] J. Wu, *Analysis on the Competitiveness of Chinese Commercial Banks Based on Factor Analysis*, Jilin University, Jilin, China, 2017.

[12] T. Liu, "Research on the application of factor Analysis model," *Journal of Physics: Conference Series*, vol. 1952, no. 4, 2021.

[13] Y. Zhu, Y. Huang, and Y. Yan, "Research on food redistribution model based on principal component analysis and factor analysis," *Journal of Physics: Conference Series*, vol. 1952, no. 4, 2021.

[14] L. Lei, *An Improved SOM Neural Network and its Application in Water Quality Evaluatio*, Chongqing University, Chongqing, China, 2009.

[15] L. Lei and W. Shi, "Fan min Application of improved SOM neural network in water quality evaluation and analysis,"

Journal of Instrumentation, vol. 30, no. 11, pp. 2379–2383, 2009.

[16] J. Yang, J. Zhan, and J. Zhang, *30 Cases of MATLAB Neural Network*, Electronic Industry Press, Beijing, China, 2014.

[17] D. Han and Y. Tang, "SOM + K-means two-stage clustering coal quality big data mining method and application," *Coal Science and Technology*, vol. 1-12, 2022.

[18] J. Niu, "Intelligent evaluation model of e-commerce transaction volume based on the combination of k-means and SOM algorithms," *International Journal of Information and Communication Technology*, vol. 18, no. 2, 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9541115, 26 pages
https://doi.org/10.1155/2022/9541115

Research Article

Jellyfish Search-Optimized Deep Learning for Compressive Strength Prediction in Images of Ready-Mixed Concrete

Jui-Sheng Chou, Stela Tjandrakusuma, and Chi-Yun Liu

National Taiwan University of Science and Technology, Taipei, Taiwan

Correspondence should be addressed to Jui-Sheng Chou; jschou@mail.ntust.edu.tw

Received 9 April 2022; Accepted 7 June 2022; Published 1 August 2022

Academic Editor: Ripon Chakrabortty

Most building structures that are built today are built from concrete, owing to its various favorable properties. Compressive strength is one of the mechanical properties of concrete that is directly related to the safety of the structures. Therefore, predicting the compressive strength can facilitate the early planning of material quality management. A series of deep learning (DL) models that suit computer vision tasks, namely the convolutional neural networks (CNNs), are used to predict the compressive strength of ready-mixed concrete. To demonstrate the efficacy of computer vision-based prediction, its effectiveness using imaging numerical data was compared with that of the deep neural networks (DNNs) technique that uses conventional numerical data. Various DL prediction models were compared and the best ones were identified with the relevant concrete datasets. The best DL models were then optimized by fine-tuning their hyperparameters using a newly developed bio-inspired metaheuristic algorithm, called jellyfish search optimizer, to enhance the accuracy and reliability. Analytical experiments indicate that the computer vision-based CNNs outperform the numerical data-based DNNs in all evaluation metrics except the training time. Thus, the bio-inspired optimization of computer vision-based convolutional neural networks is potentially a promising approach to predict the compressive strength of ready-mixed concrete.

1. Introduction

Structures like buildings, bridges, highways, and dams are currently built using concrete as their construction material, owing to its numerous advantages, such as strength, durability, and versatility. Its compression capacity, adaptability, and resistance to climate-induced erosion and corrosion make concrete one of the best construction materials. Compressive strength is one of the principal mechanical properties of concrete that is directly related to the safety of the structures that are built from it. The compressive strength of concrete must comply with relevant standard codes, which vary among countries.

To determine the compressive strength of concrete, a cubic or cylindrical sample is typically tested using a compressive testing machine after the required curing time. These tests are labor-intensive and time-consuming. Methods such as regression methods and numerical simulation have been proposed to solve this problem and to predict the compressive strength of concrete. However, the complex nonlinear correlation between relevant variables makes the obtaining of accurate values of compressive strength very difficult.

With the advances of artificial intelligence (AI) and increases in computing power [1, 2], deep learning (DL) is being applied in an increasing number of fields. DL, which is a form of AI, has been shown to be effective in making more accurate predictions than conventional methods in many situations. One DL technique, computer vision, is often used to extract information from visual media, such as images and videos. Used in various fields, computer vision-based technique is effective for image classification, object detection, and semantic segmentation.

Several studies of the prediction of concrete compressive strength have involved the use of DL techniques [3] to improve model performance, but few have involved image recognition. The latest study of the use of image recognition to determine compressive strength had good results, but it used only 74 sets of concrete data and a single-layer convolutional neural network [4]. To examine and improve the

effectiveness of image recognition, in this study, a large dataset of ready-mixed concrete is used with convolutional neural networks (CNNs) that involve a prediction model with deep layers to extract high-level features from inputs.

Model accuracy is often evaluated with the use of a cross-fold validation or random split method to partition the source data for the testing of the training model [5]. Such methods are often called into question as overfitting occurs owing to the information leakage within the original dataset in the training process. Therefore, when putting the model into practice, it often shows a relatively poor forecast performance. Because the concrete data are accumulated over time in ready-mixed plants, the built model should be tested via a latest dataset to reflect its reasonable prediction accuracy in future use.

In this investigation, the effectiveness of the computer vision-based approach in predicting the compressive strength of ready-mixed concrete by converting numerical data to images is tested. In much research, the prediction of the compressive strength of concrete uses numerical data as inputs. The results thus obtained using those computer vision-based techniques are compared with those obtained using numerical data. With this logic, a collection of numerical values that are represented as images are the inputs to a DL technique that uses CNN-based models, which have been shown to provide accurate image classification in the domain of computer vision.

The effectiveness of the computer vision-based technique was tested by comparing the results with those of another DL technique that uses deep neural networks (DNNs) with numerical data for model construction. To maximize the accuracy, a metaheuristic optimization algorithm was used to finetune the hyperparameters of the best DL models. Instead of using the cross-fold validation or random split method within the original dataset, a newly collected dataset in the upcoming year was used for the testing of the training model. This approach meets the practical needs and operations in estimating compressive strength at ready-mixed concrete plants.

This paper is organized as follows. Section 2 reviews the relevant literature. Section 3 describes the methodology and performance metrics that are used herein. Section 4 presents the collection and preprocessing of data, implementation of the DL models, the experimental results obtained using the optimized DL models, and sensitivity analysis of modeling performance. The final section summarizes the findings and limitations of the method and makes recommendations for future studies.

2. Literature Review

2.1. Conventional Compressive Strength Prediction of Ready-Mixed Concrete. Ready-mixed concrete is typically manufactured in a concrete plant before being transported to a construction site. In a concrete plant, ready-mixed concrete is manufactured by combining several raw materials with a specific design mix ratio to create concrete with certain desirable properties. Figure 1 presents the manufacturing process of ready-mixed concrete.

The compressive strength of concrete is commonly tested using a compression test machine, which performs a mechanical test to measure the maximum compressive load that can be borne by a concrete sample [6]. Before testing, the sample must be cured for a specified curing period. Nondestructive tests (such as ultrasonic or pulse velocity tests [7] and conductivity tests [8]) have also been proposed to determine the compressive strength due to the lack of correlation between the standard compression test value with the real strength of concrete in a structure. These tests, however, have disadvantages with respect to time, cost, and labor.

Owing to the disadvantages of mechanical tests, empirical models [9, 10] for calculating the compressive strength of concrete have been developed. Empirical methods (e.g., multiple linear regression), however, have been shown to be somewhat ineffective for calculating the compressive strength of concrete because of the nonlinear behavior in relevant concrete variables. The compressive strength of concrete is influenced by numerous factors, as it is formed by complex reactions among concrete materials (such as cement and aggregates) and the environment (as in curing) [11].

2.2. Deep Learning to Determine Concrete Compressive Strength. In recent years, the field of artificial intelligence (AI) has grown very rapidly. AI methods are used in a wide variety of fields, including seismology [12], energy systems [13], and civil engineering [14]. In several studies, AI has been used to determine the concrete compressive strength, using real data for concrete to build a prediction model. During the training of the prediction model, various composite materials of concrete, such as cement, water, sand, and gravel, are used as predictors to yield a model that best fits the given training data. After validation, the model is then used to predict the compressive strength.

An advanced branch of AI, deep learning (DL), has performed excellently in fields such as computer vision [15]. Many studies [16–18] have shown that DL exhibits outstanding prediction performance, especially in image and video recognition. In this field, the commonly used DL techniques include those based on convolutional neural networks (CNNs) [19]. A recent study confirmed that the CNN model (Visual Geometry Group, VGG) achieved a 98% accuracy in concrete compressive strength prediction, which was 2% and 12% greater than the machine learning models, random forest (RF) and support vector regression (SVR), respectively [20].

2.3. Hyperparameter Optimization with Metaheuristic Algorithm. In the training of the DL models, additional optimizers are often required, as the models have several hyperparameters (such as the epsilon of batch normalization, batch size, epoch, learning rate, and dropout rate) that influence their predictive performance [21, 22]. To find the values of hyperparameters that yield the best prediction model, optimization algorithms (such as the greedy algorithm [23]) that are based on iterative methods (such as

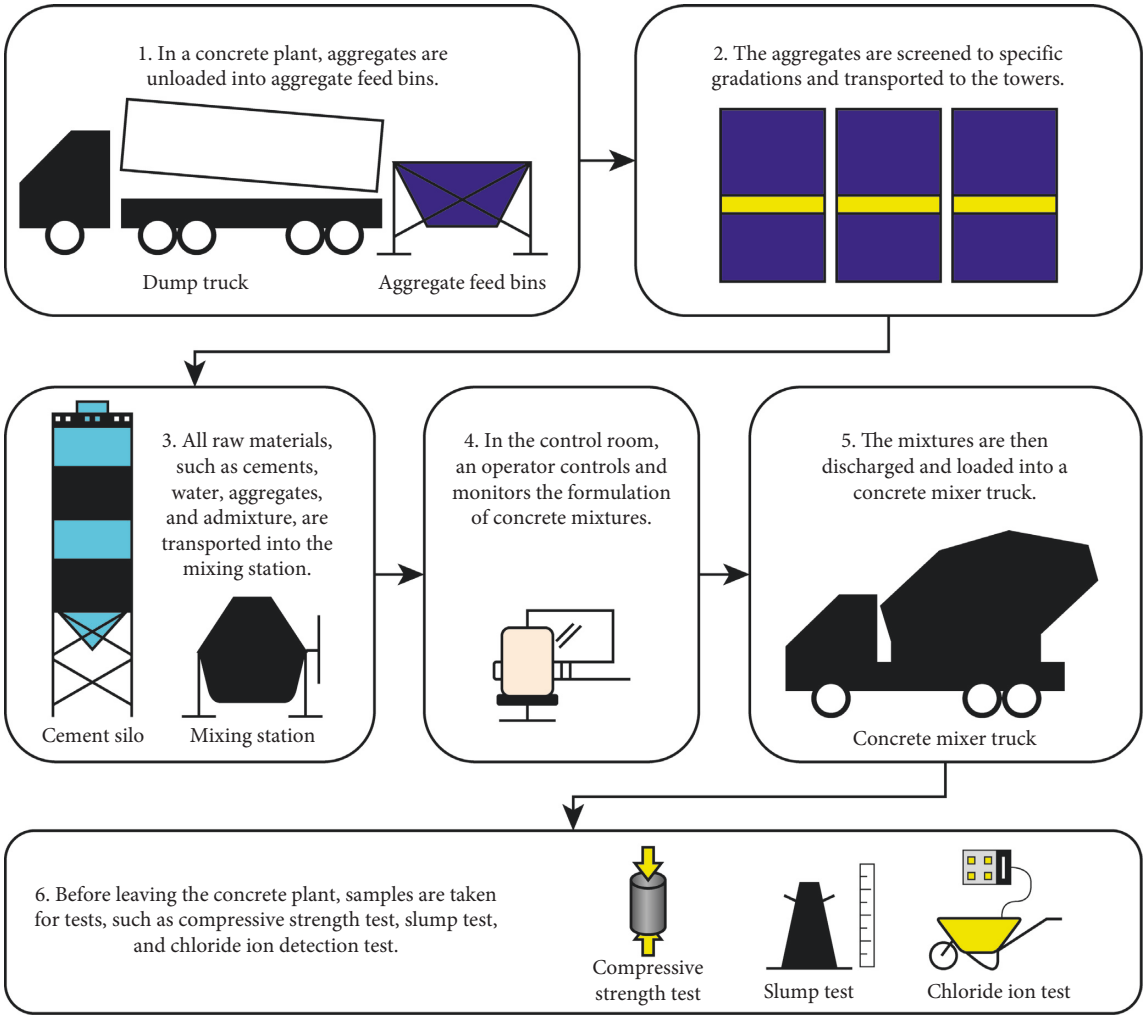


Figure 1: Ready-mixed concrete manufacturing process.

gradient descent [24]) or heuristic methods [25] are often used. However, such methods may not always lead to the optimal solution and consume a significant computational time compared to modern metaheuristic algorithms.

The metaheuristic algorithm, with its ease of implementation and effectiveness in various fields, is becoming increasingly popular for use in solving optimization problems. Recently, several newly developed metaheuristic optimizers have outperformed the well-known metaheuristic algorithms [26, 27]. The Jellyfish Search (JS) algorithm [27], in particular, has great efficacy because it requires little tuning of algorithm-specific parameters. Consequently, the JS algorithm was used in this study to optimize the DL models.

3. Methodology

3.1. Deep Learning and Computer Vision-Based Techniques

3.1.1. Deep Neural Networks. Artificial neural networks (ANNs) consist of information processing units that are arranged in layers similar to neurons in the human brain. An ANN typically comprises layers of three types: an input layer, hidden layers, and an output layer. The architecture that is used in a deep learning model typically consists of more than four hidden layers. Figure 2 displays a simple ANN model architecture.

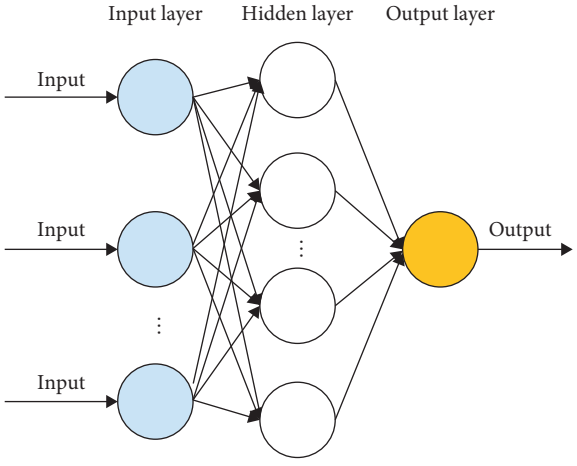


Figure 2: Simple ANN model architecture.

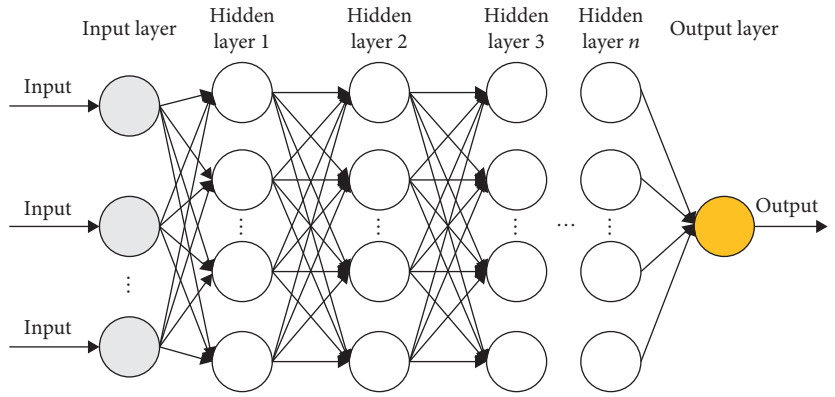


Figure 3: Deep neural network (DNN) model architecture.

An input layer receives data and an output layer generates a prediction. In the hidden layers, inputs are processed and the information that is obtained from the processes is passed to the next layer. Values from the input layer are transformed by multiplying them by weights and adding bias values.

Several types of ANN vary in implementation. A fully connected neural network is an ANN that consists of connected neurons. In such an ANN, all neurons in a layer are connected to the neurons in the next layer. Likewise, standard feedforward neural networks (FNNs) consist of numerous connected neurons, and each connection transmits information to other neurons in one forward direction [28].

Notably, internal hyperparameters affect the learning of an ANN model. A hyperparameter is a constant parameter that is set before the training begins. Some examples of hyperparameters in ANNs are the number of hidden layers, learning rate, batch size, and epoch. In contrast, parameters such as weights and bias values change throughout the learning process.

A deep neural network (DNN) is a neural network that differs from a typical ANN with respect to architecture. DNNs have multiple hidden layers (Figure 3) that are used to extract high-level features from the input data. Additional layers typically correspond to additional parameters (such as weights and biases) in a model. Accordingly, DNNs can capture complex nonlinear relationships [28].

3.1.2. Convolutional Neural Network-Based Models. A convolutional neural network (CNN or ConvNet) is a connected neural network that is generally effective for solving computer vision problems, such as image feature extraction, classification, object detection, and semantic segmentation. A CNN commonly learns patterns by processing image or video data. It can detect objects, identify the locations of the objects, and differentiate or segment them inside an image.

A generic CNN usually comprises an input layer, multiple hidden layers, which include convolutional layers, pooling layers, fully connected layers, and dropout layers, and an output layer (Figure 4). In the input layer, the model

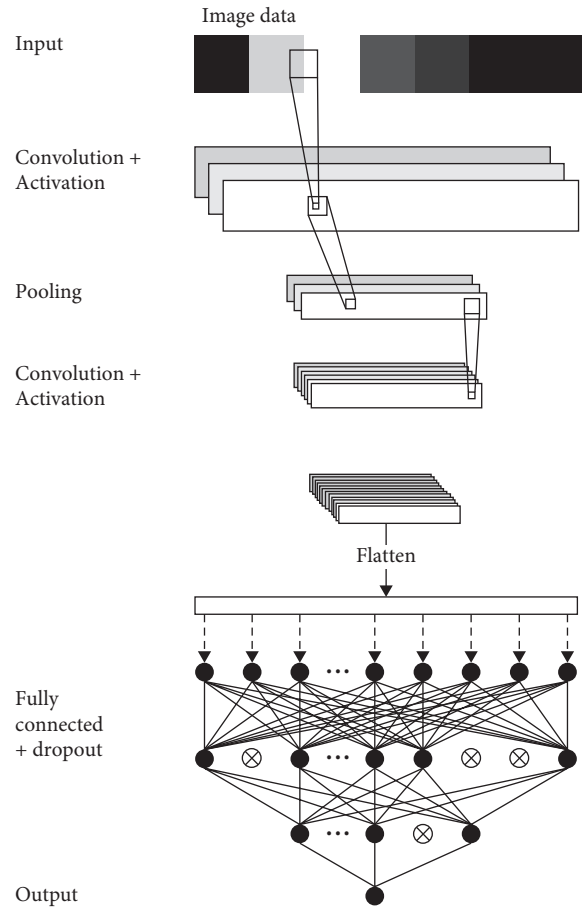


Figure 4: Generic convolutional neural network (CNN) model architecture.

receives images as inputs and creates input tensors that contain the pixel values of the images. Input matrixes of dimensions $w \times h \times c$ are then fed to the hidden layer, where w represents the width of the image, h represents the height of the image, and c represents the number of channels. A standard colored image typically has three channels for red, green, and blue.

The convolutional layer in the CNN model processes the previous matrixes into smaller forms without losing any

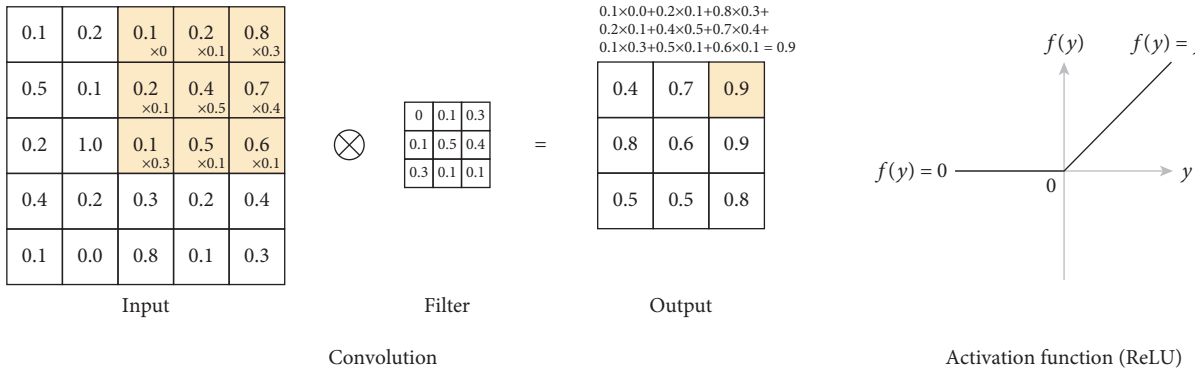


Figure 5: Convolutional layer multiplication process and plot of ReLU.

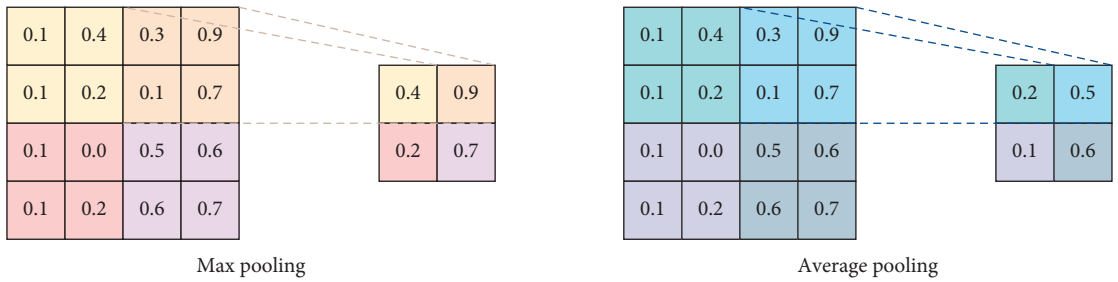


Figure 6: Example of max pooling and average pooling.

feature by generating weight values of a filter or kernel of a certain size ($m \times m$) and then multiplying the filter ($n \times n$) by the input matrixes. Convolution operation is defined as follows [29]:

$$C = I \otimes F. \tag{1}$$

Here, I is the input image data; F is the filter; $\otimes$ denotes the convolution operation; C is the convolution map of size $(o \times o)$, in which $o = m - n + 2zp/s + 1$; s is the stride and denotes the number of pixels by which F is sliding over I; and zp is the zero padding. Usually, it is necessary to add a bounding of zeros around I to preserve complete image information. The values thus obtained are summed (Figure 5). Sliding over all parts of input matrixes, the convolutional layer generates, as an output, a new feature map of certain features in the image.

After the multiplication processes, a CNN model typically applies an activation function that introduces nonlinearity to the model to help it learn complex patterns in the data. A general form of activation function is defined as follows:

$$C_m = f(C). \tag{2}$$

C_m is the convolution map after applying the nonlinear activation function f. Of the many available activation functions, the rectified linear unit (ReLU) is commonly used, as it provides better training results than other activation functions [30]. A ReLU function is a simple calculation that returns the original input values or sets the value to zero if the input is less than or equal to zero (Figure 5).

The pooling layer in the model reduces the size of the input matrixes by reducing the number of parameters and the amount of computation in the network, preventing overfitting. Similar to a convolutional layer, a pooling layer takes several input values inside a filter from the previous layer and the filter is shifted over some pixels at a time until all parts of the input matrix are processed. Common pooling layer types are average pooling or max pooling (Figure 6). The pooling operation also called downsampling operation is expressed as follows:

$$P_m = P_o(C_m), \tag{3}$$

where P_m is the pooling map and P_o is the pooling operation.

After the operation of several convolutional layers and pooling layers, a CNN model typically flattens the output matrix of the previous layer into a single vector of values. The single vector of values is input to a fully connected layer to extract the features that were learned in the previous layers and to classify the input images. In this layer, the probabilities that an object in the input image is a member of the possible classes are calculated. The model output of the i^{th} fully connected hidden layer is expressed as follows [29]:

$$Y^i = f\left(H^i\right), \tag{4}$$

where the weight sum vector H^i is

$$H^i = w^i Y^{i-1} + B^i. \tag{5}$$

w is the connected weight of the artificial neurons. f is a nonlinear activation function (e.g., sigmoid, Tanh, ReLU, etc.). The bias value B^i defines the activation level of the artificial neurons.

In neural networks, when the parameters of a layer change, so do the distribution of inputs to subsequent layers.

These shifts in input distributions can be problematic for neural networks. To alleviate this concern, many normalization operations, such as Batch Normalization (BN), Layer Normalization (LN), and Instance Normalization (IN), have been proposed. For example, given an input batch of height h and width w with n samples and c channels $x \in R^{n\times c\times h\times w}$, BN normalizes the mean and standard deviation for each individual feature channel during training [31].

$$BN(x) = \gamma\left(\frac{x-\mu_B}{\sigma_B}\right) + \beta, \tag{6}$$

where $\gamma, \beta \in R^c$ are referred to as the scale and the shift parameters for the channel; $\mu_B, \sigma_B \in R^c$ are the mean and standard deviation, respectively, computed across batch size and spatial dimensions independently for each feature channel.

Adding a dropout layer is an effective regularization technique to improve the generalization capability and mitigate overfitting of models. Dropout function can be formulated as follows [32]:

$$\tilde{f}^l(x_i) = f^l(x_i) - m_i^l * f^l(x_i), \tag{7}$$

where $*$ denotes the element-wise product and $f^l(x_i)$ and $\tilde{f}^l(x_i)$ are the original feature and distorted features, respectively. In addition, $\in m_i^l\{0,1\}^{d^l}$ is the binary mask applied on feature map $f^l(x_i)$ in which d^l is the dimension of the feature map of l-th layer, and each element in m_i^l is drawn from Bernoulli distribution and set to 1 with the dropping probability. Undoubtedly, implementing dropout on the features in the training phase will force the given network paying more attention on those non-zero regions, and partially solve the overfitting.

In this decade, various CNN models and their advanced variants have been developed. Some common and popular CNN models are VGG [33], residual neural networks (ResNets) [34, 35], Inception [36, 37], extreme inception (Xception) [38], MobileNet [39, 40], DenseNet [41], NASNet [42], and EfficientNet [43]. These CNN-oriented models have different architectures, which are briefly introduced as follows:

(1) VGG. VGG [33] uses a very small kernel (3×3) rather than one of a previously common size, 5×5 or 7×7, which would have a wider scanning area. The small kernel is used uniformly throughout all layers. Although the overall architecture is simple, the VGG has an enormous number of parameters. Figure 7 displays the architectures of two common VGG models, VGG16 and VGG19, which comprise 16 and 19 deep layers, respectively. In the figure, the convolutional layer is denoted as "<kernel size> Conv, <filter>."

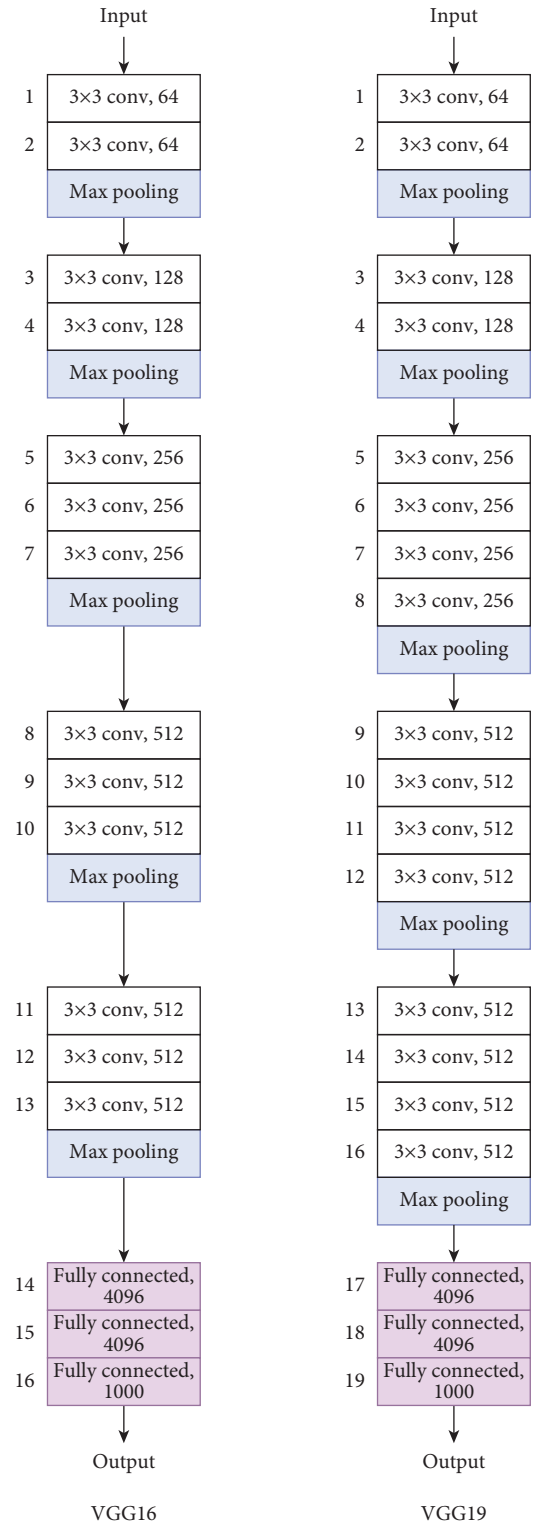


Figure 7: VGG16 and VGG19 models' architectures.

(2) ResNet. Increasing the depth of a CNN by adding layers to its architecture up to a certain limit should help the corresponding CNN model to learn more complex features, but a vanishing gradient problem typically prevents the effective training of a CNN model in many-layered networks. A vanishing gradient problem can prevent the weights in the network from being updated, potentially stopping the training of the CNN model. To solve this problem in residual neural networks (ResNets), the network implements "residual connections" or "skip connections."

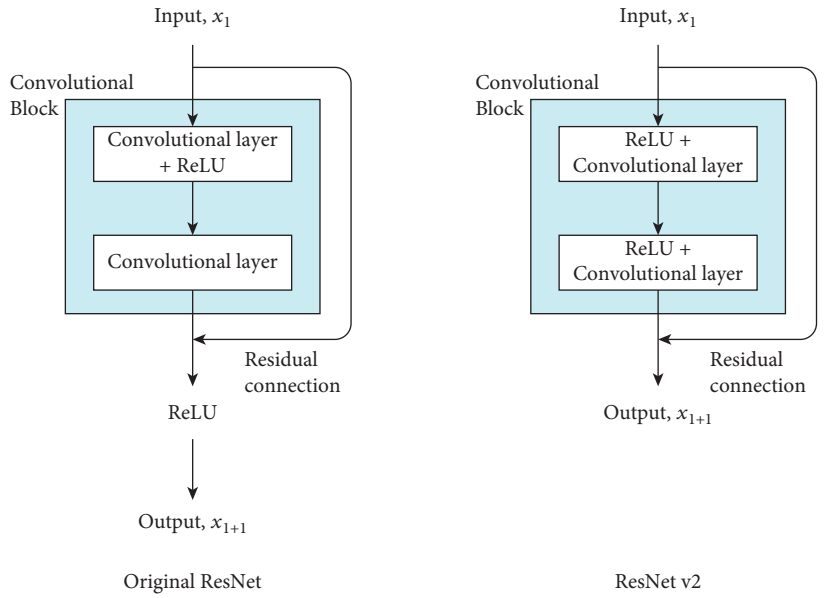


Figure 8: Residual connection.

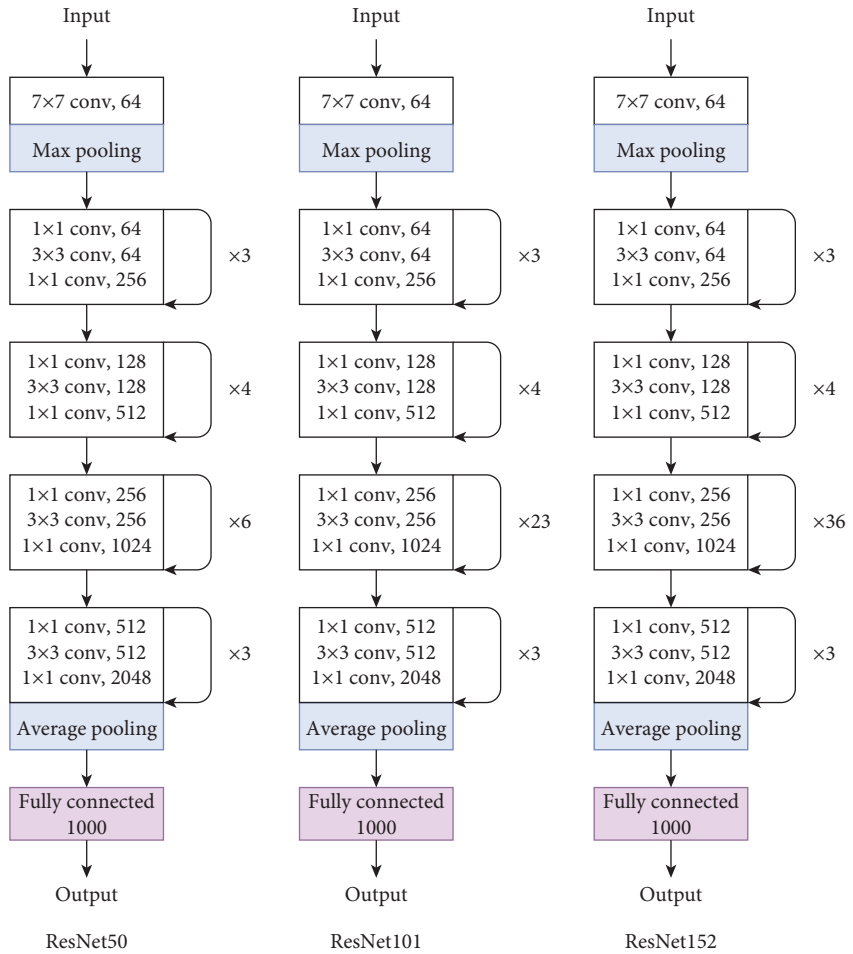


Figure 9: ResNet model architecture.

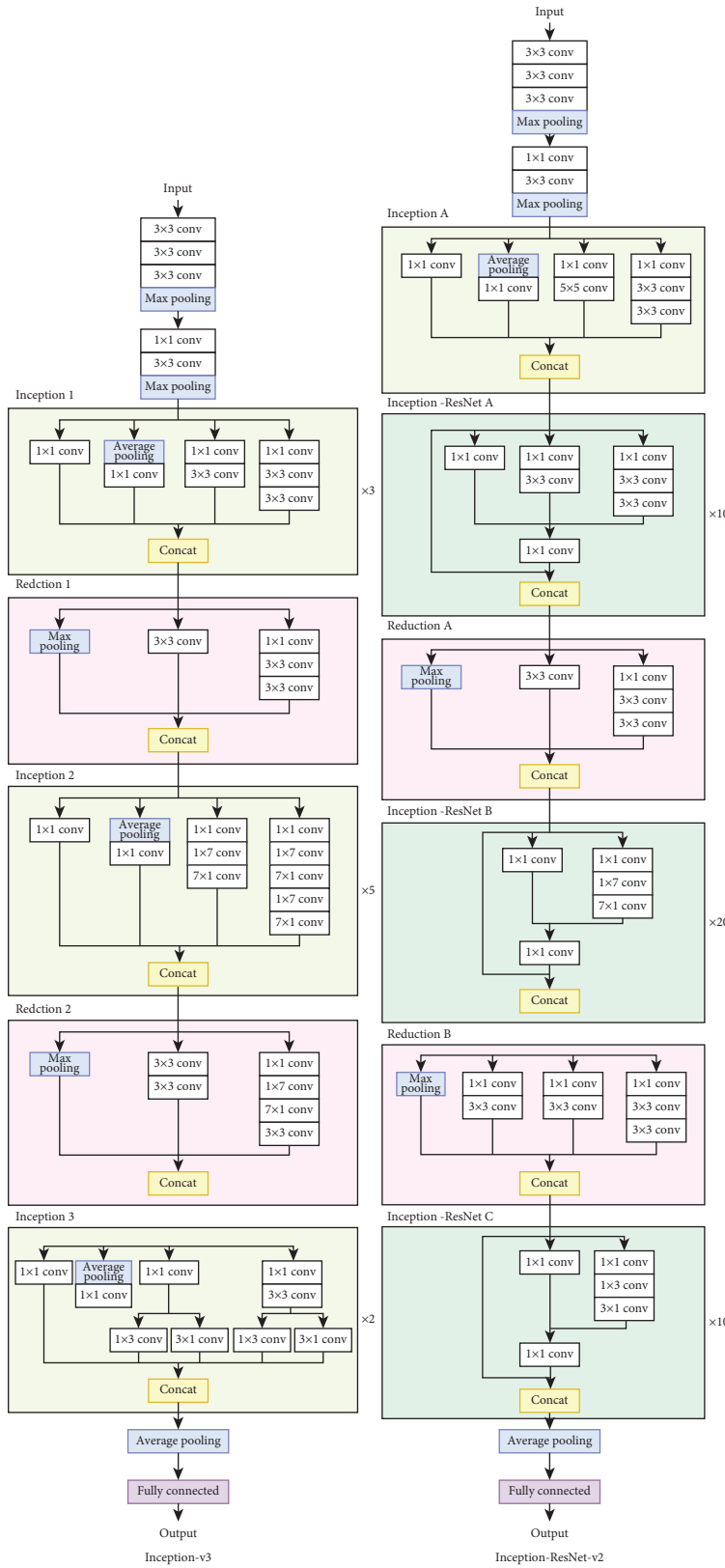


Figure 10: Inception-v3 and Inception-ResNet-v2 models' architectures.

A residual connection refers to a shortcut connection that is added inside a CNN architecture to allow information to be passed or added through layers of the convolutional block (Figure 8). In the original ResNet, a shortcut connection is added before the activation function is implemented, while in ResNet v2 [34], activation functions are

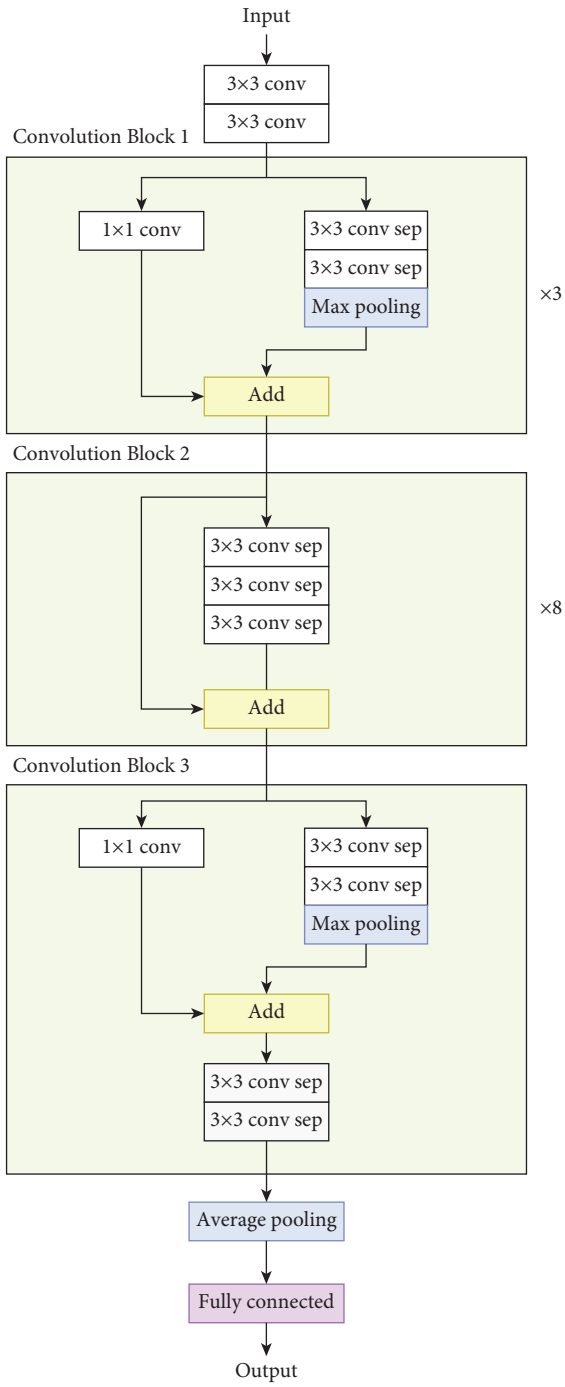


Figure 11: Xception model architecture.

implemented before the convolutional layer and the shortcut connection is added after. Figure 9 presents the architectures of ResNet50, ResNet101, and ResNet152, which comprise 50, 101, and 152 deep layers, respectively.

(3) Inception. Inception architecture [36] is the first CNN model architecture that exhibits the advantages of branching a convolutional path into multiple paths. In Inception, the CNN model uses filters of various sizes in various paths. At the end of the block, the model concatenates the outputs of

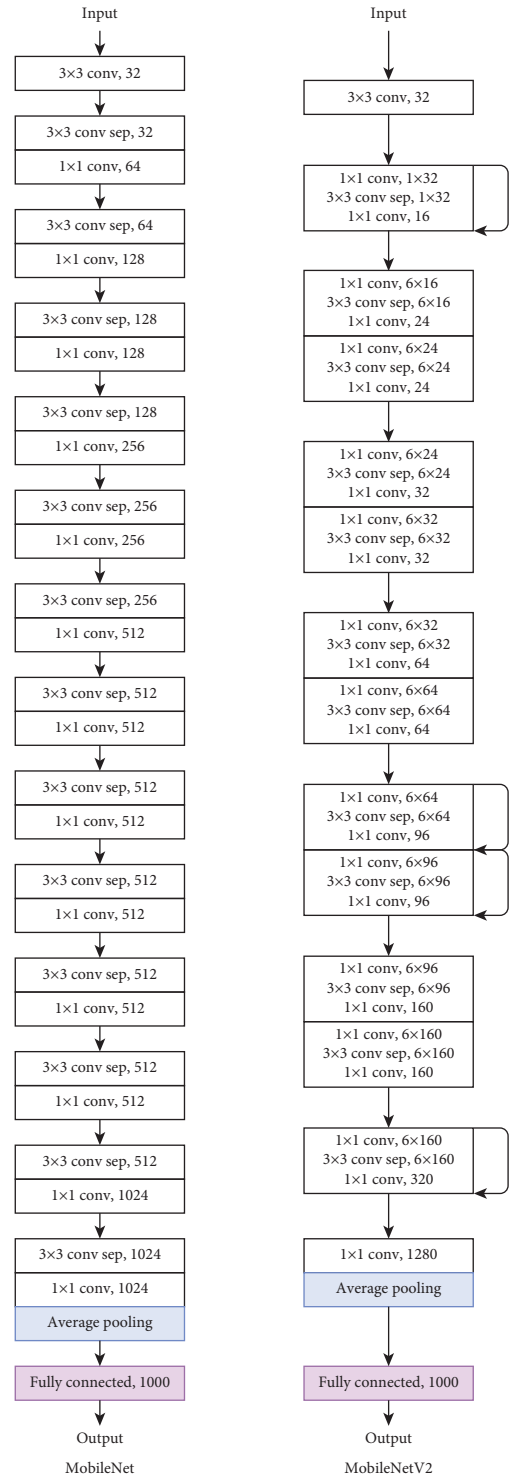


Figure 12: MobileNet and MobileNetV2 models' architectures.

the paths. In Inception-v3 [36], the Inception model is improved by changing the original 5×5 and 7×7 convolution kernels to two 3×3 and three 3×3 convolutional kernels, respectively. These changes in the architecture help the model reduce the amount of computation that is required during the training process.

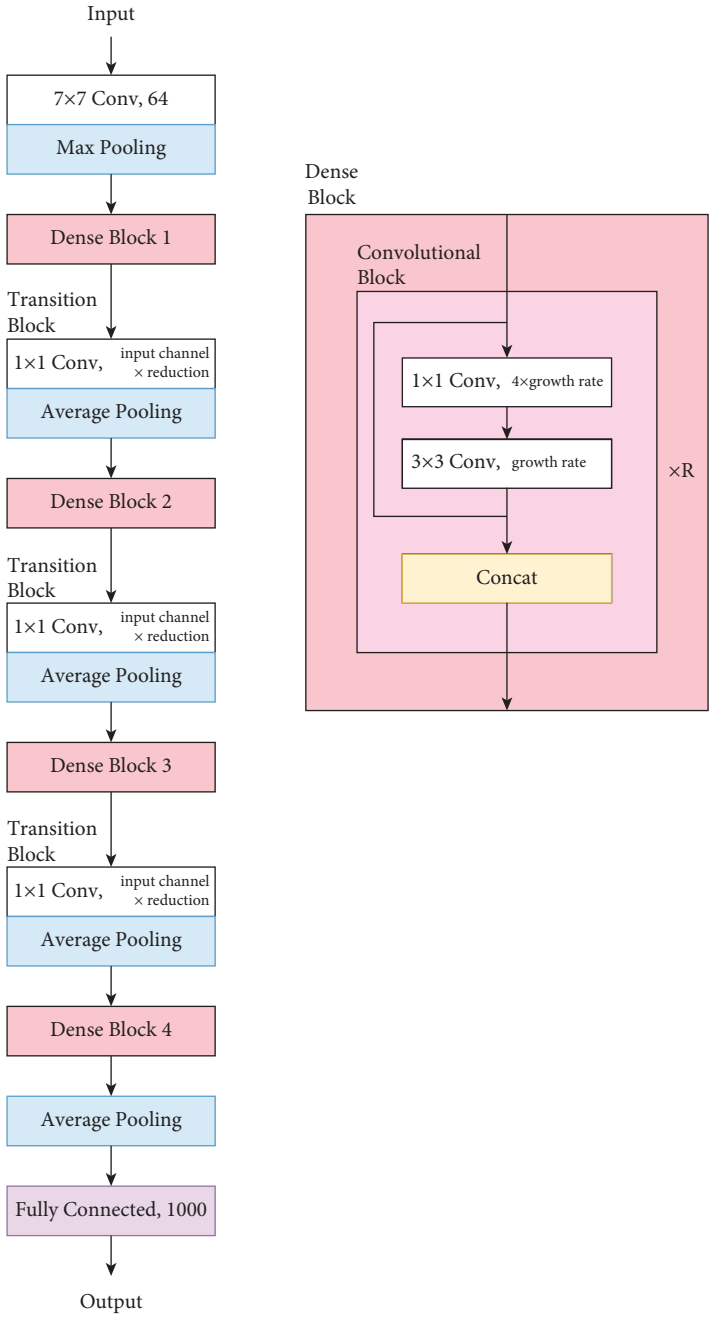


Figure 13: DenseNet model architecture.

In Inception-ResNet-v1 [37] and Inception-ResNet-v2 [37], the original inception blocks are converted into residual inception blocks. The Inception-ResNet-v2 model differs from the Inception-ResNet-v1 model in that it is more computationally burdensome. However, it outperforms the original Inception and ResNet models. Figure 10 displays the Inception-v3 and Inception-ResNet-v2 models' architectures.

(4) Xception. The Xception (or Extreme Inception) [38] architecture (Figure 11) is inspired by the Inception model. In Xception, the original inception blocks are replaced by depthwise separable convolutions. A depthwise separable convolution consists of a depthwise convolution and a 1×1 convolution. A depthwise convolution is a spatial convolution that performs convolutional multiplications independently over each channel. In depthwise convolution, a convolutional kernel only iterates one channel of the input, not all channels.

(5) MobileNets. MobileNets [39] refer to a type of CNN model whose objectives are to reduce the number of parameters and the number of computations while maintaining accuracy. Accordingly, MobileNets use depthwise

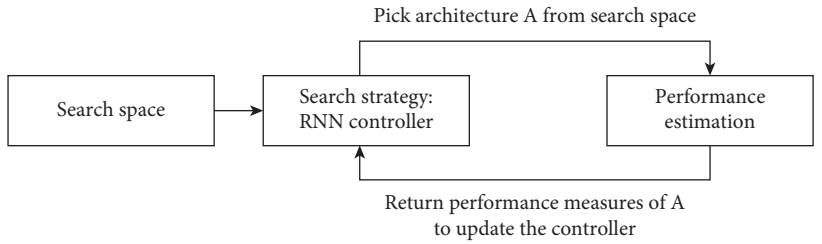


Figure 14: Neural architecture search method.

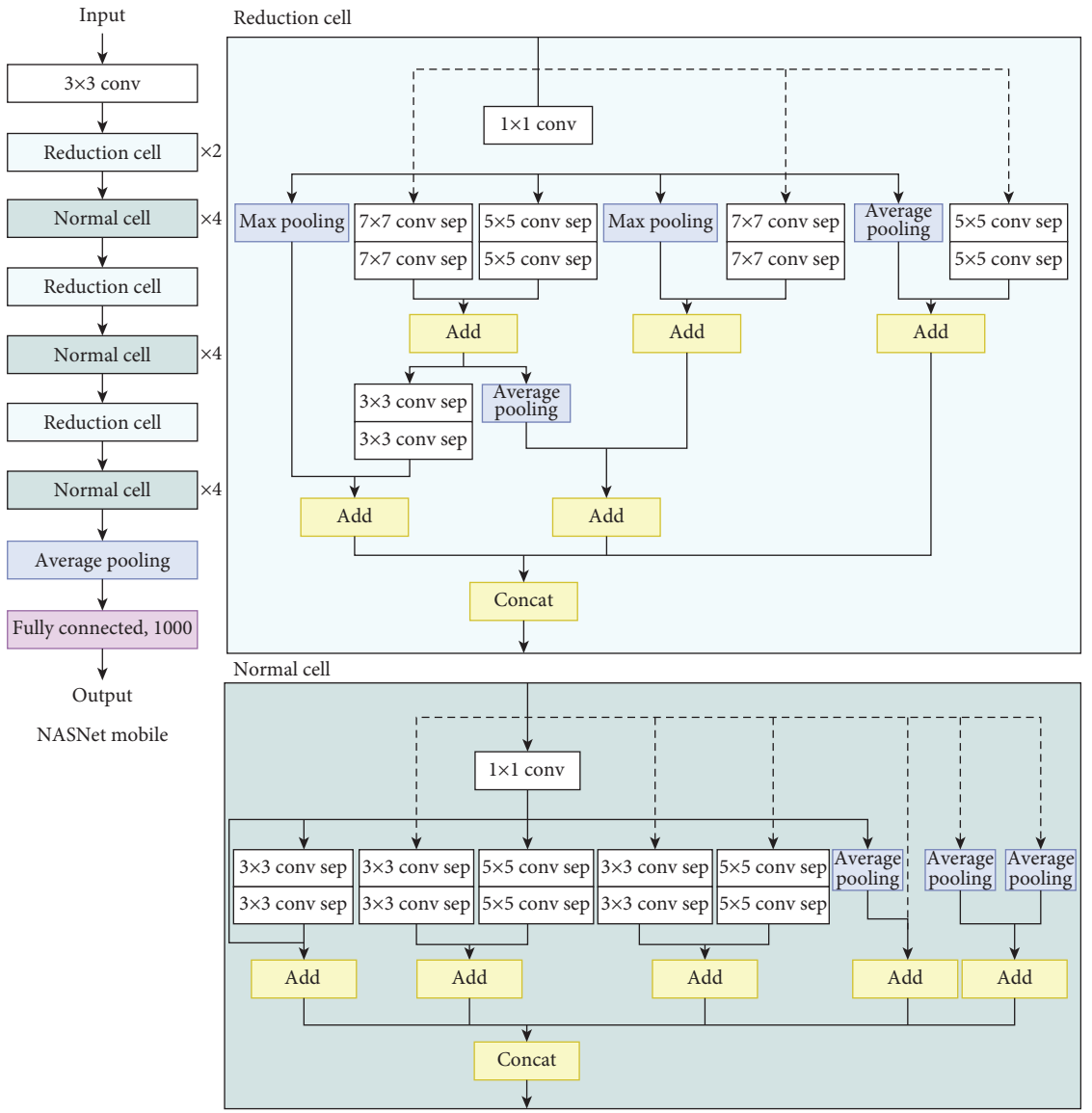


Figure 15: NASNet-A mobile model architecture.

separable convolutions. They are typically used in mobile devices or embedded applications, and so have a small architecture. In MobileNets, width multiplier and resolution multiplier hyperparameters are implemented to thin the network and to rescale the input image, respectively.

Similar to the original MobileNet, MobileNetV2 [40] is built for mobile devices. In MobileNetV2, an inverted residual structure, which consists of linear bottleneck layers, is used. An inverted residual structure expands a low-dimensional feature map to a high-dimensional one, uses

depthwise convolutions, and projects back features to a low-dimensional representation using a linear convolution. MobileNetV2 has fewer parameters than the original MobileNet. Figure 12 displays the original MobileNet and MobileNetV2 architectures.

(6) DenseNet. The main intent of a dense convolutional network (DenseNet) [41] is to use short connections between layers by connecting the network layers to every other layer in the forward direction. Therefore, the inputs of each network layer include the feature maps of all preceding layers. This approach has been shown to improve the accuracy of a CNN. Figure 13 displays the DenseNet architecture.

(7) NASNet. The neural architecture search network (NASNet) [42] is used to solve the problem of finding a good CNN architecture by finding a neural network architecture or the best combination of parameters in a CNN with a recurrent neural network (RNN) acting as a controller. Figure 14 presents the neural architecture search method that is used in a NASNet model. Figure 15 displays one of the model architectures, NASNet-A, for the mobile version, which is found using the neural architecture search method.

(8) EfficientNet. EfficientNet is a type of CNN model that uniformly scales all depth, width, and resolution dimensions using a compound scaling coefficient. A total of eight CNN models are developed based on this idea. The models are named EfficientNets followed by B0, B1, B2, B3, B4, B5, B6, and B7. The EfficientNet architecture includes a total of seven network blocks (Figure 16). The number of subblocks inside varies with the EfficientNet models that are used [43].

3.2. Metaheuristic Optimization Algorithm: Jellyfish Search Optimizer. One of the challenges that is associated with the deep learning models is the finding of optimal hyperparameters. To solve this hyperparameter optimization problem, a metaheuristic optimization algorithm is frequently used. Considerable research has been done on the development of metaheuristic algorithms, and some of them have become well known for their effectiveness in solving optimization problems [44–46]. The metaheuristic algorithms primarily vary in the balance between their two main phases—exploration and exploitation [47].

A newly developed metaheuristic optimization algorithm, the Jellyfish Search (JS) optimizer [27], has considerably outperformed many other well-known metaheuristic optimization algorithms and it requires less algorithm-specific parameter tuning than some well-known metaheuristic algorithms. The optimizer requires the setting of only two controlling parameters, which are the number of iterations and population size. In a JS optimizer, the population of jellyfish is initialized using a logistic map, which generates varying initial populations.

Since the optimization algorithm is inspired by the behavior of jellyfish as they search for food in the ocean, the objective function of the JS optimizer is the location of jellyfish where it has the most food. In a JS optimizer, the exploration phase involves the movement of jellyfish as they follow ocean currents in search of food, while the exploitation phase involves the passive and active motions of the jellyfish inside a jellyfish swarm. Figure 17 presents the six phases of jellyfish in the ocean [27], including phase 1: jellyfish in the ocean; phase 2: following the ocean current; phases 3–5: passive and active motions inside the jellyfish swarm that are switched to each other according to a time control mechanism; and phase 6: reach the jellyfish bloom.

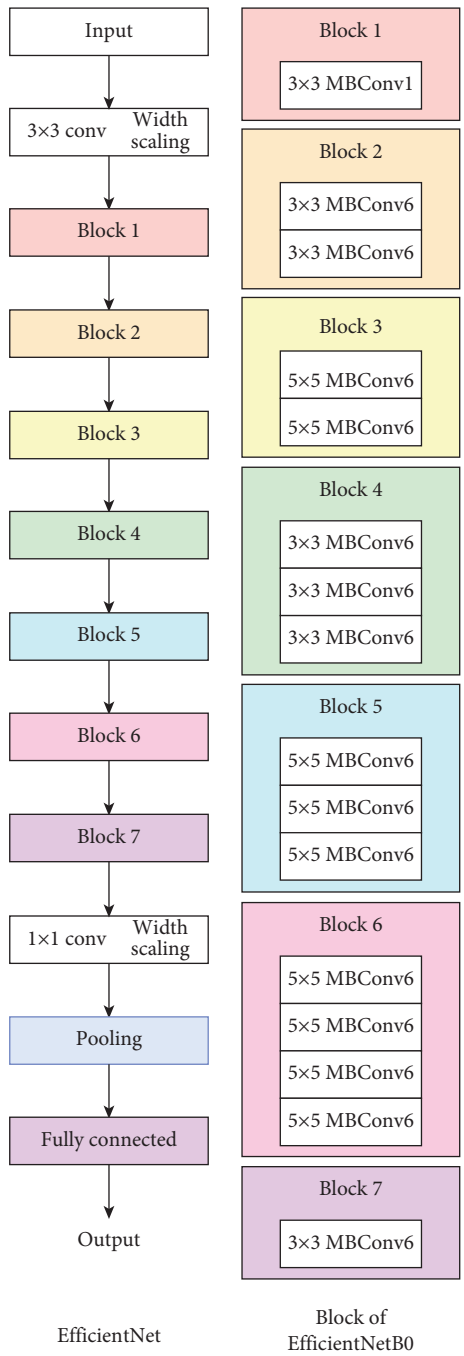


Figure 16: Efficientnet model architecture.

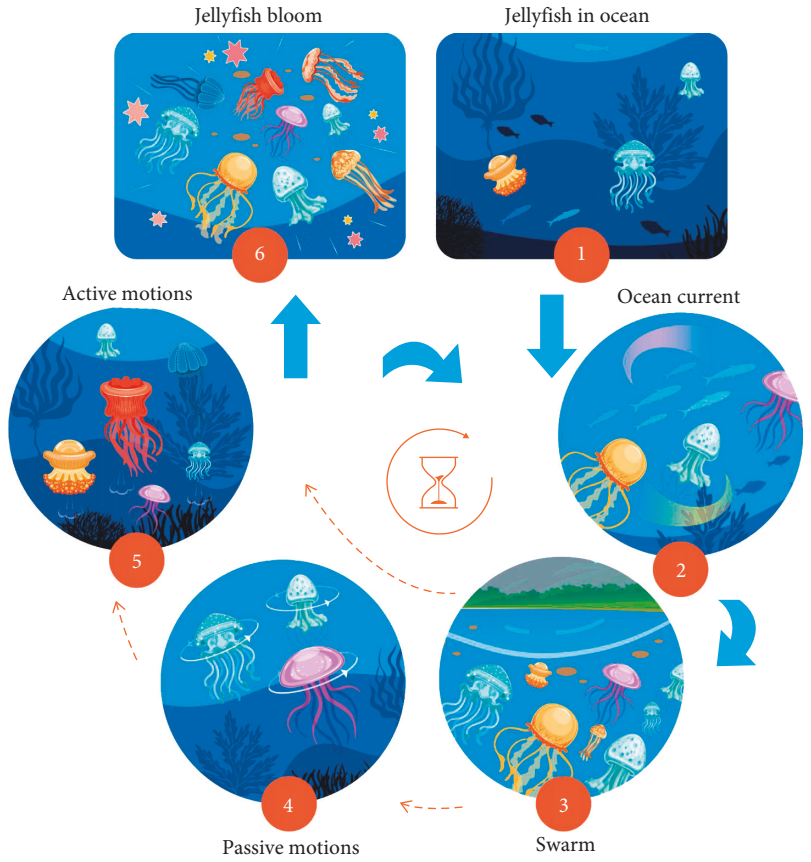


Figure 17: Phases of the jellyfish search algorithm.

3.2.1. Movement Following Ocean Current. Ocean currents carry a large amount of food, attracting jellyfish to them, and thus jellyfish follow them. The following equation represents the direction of the ocean current, ($\overrightarrow{trend}$), and the new location of a jellyfish after it moves, $X_i(t+1)$ [27].

$$\begin{aligned}\overrightarrow{\text{trend}} &= X^* - 3 \times \text{rand}(0,1) \times \mu,\\ X_i(t+1) &= X_i(t) + \text{rand}(0,1) \times \overrightarrow{\text{trend}}.\end{aligned} \tag{8}$$

Here, X^* is the jellyfish at the best location, μ is the average location of all jellyfish, $X_i(t)$ are the current locations of the jellyfish at time t, and $X_i(t+1)$ are the updated locations of the jellyfish at time (t+1).

3.2.2. Motions Inside Jellyfish Swarm. The motions of jellyfish in a swarm can be grouped into passive motion (type A) and active motion (type B). Passive motion signifies a movement of a jellyfish around its original position, and active motion signifies its movement to another position. Initially, most jellyfish exhibit type A motion, but after some time, more jellyfish exhibit type B motion [27]. The new location of a jellyfish that exhibits A motion is formulated as follows:

$$X_i(t+1) = X_i(t) + 0.1 \times \text{rand}(0,1) \times (Ub - Lb), \tag{9}$$

where rand$(0,1)$ is a random number between 0 and 1, Ub is the upper bound on the search space, and Lb is the lower bound on the search space.

For type B motion, one other jellyfish, X_j, is randomly selected for use in determining the new location of the jellyfish of interest, X_i. If the amount of food at the location of X_j exceeds that at the location of X_i, then X_i will move toward X_j. Otherwise, X_i will move away from X_j. The direction of type B motion ($\overrightarrow{\text{Direction}}$) and the updated jellyfish location are given by the following equations for minimization problems:

$$\begin{aligned}\overrightarrow{\text{Direction}} &= \begin{cases} X_j(t) - X_i(t), & \text{if } f(X_i) \geq f(X_j),\\ X_i(t) - X_j(t), & \text{if } f(X_i) < f(X_j),\end{cases}\\ X_i(t+1) &= X_i(t) + \text{rand}(0,1) \times \overrightarrow{\text{Direction}},\end{aligned} \tag{10}$$

where $f(X_i)$ and $f(X_j)$ denote the objective functions at locations X_i and X_j, respectively.

3.2.3. Time Control Mechanism. A time control mechanism in a JS optimizer determines the type of jellyfish motion and controls the switching between the phases of the JS optimizer (following an ocean current and moving inside a jellyfish swarm). The equation below provides the time control function, $c(t)$.

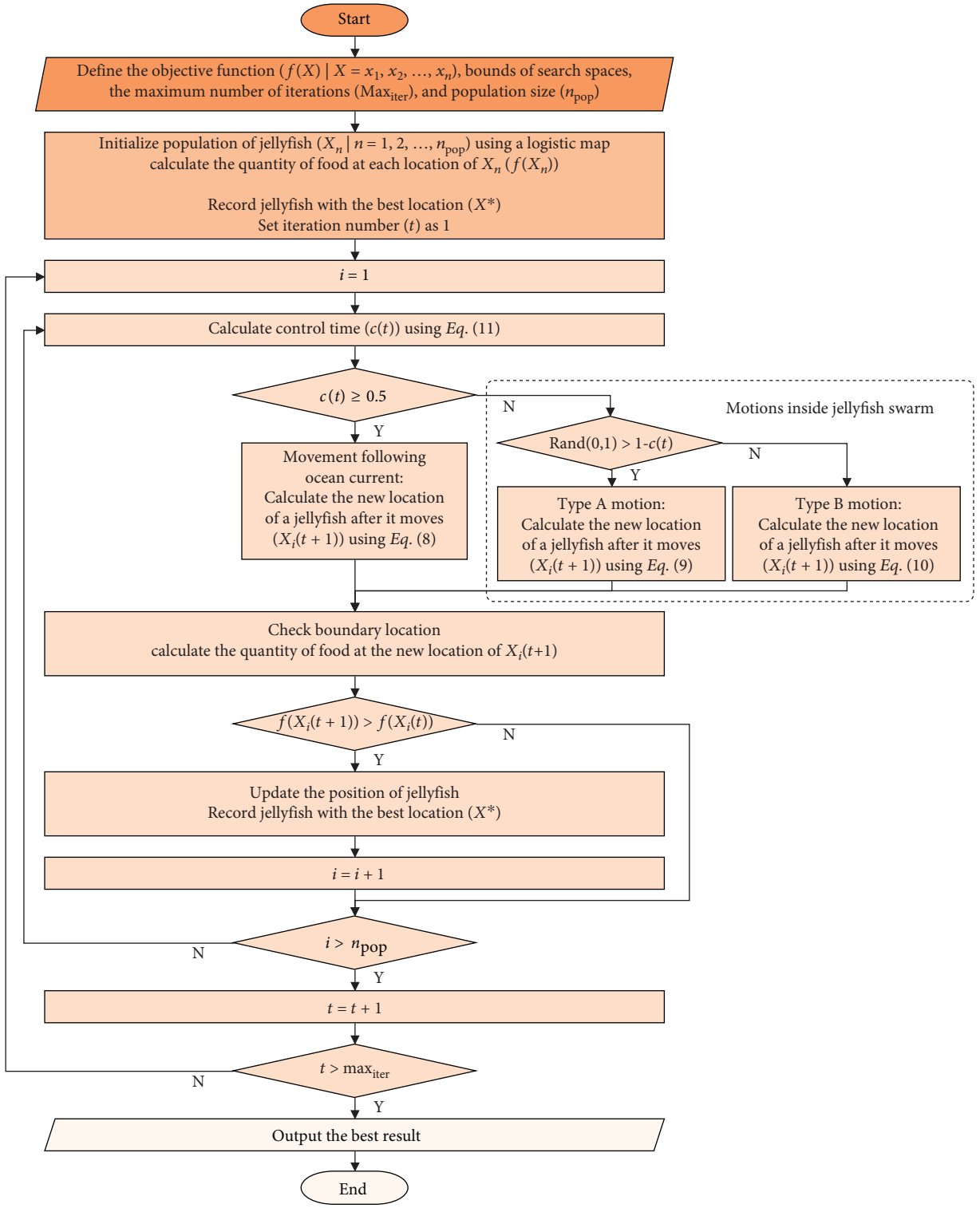


Figure 18: Algorithmic flowchart of the jellyfish search algorithm.

$$c(t) = \left| \left(1 - \frac{t}{Max_{iter}}\right) \times (2 \times \text{rand}(0,1) - 1) \right|. \quad (11)$$

Here, t is the time specified as the iteration number and Max_{iter} is the maximum number of iterations.

If the value of $c(t)$ exceeds 0.5, then the jellyfish will follow the ocean current; if it is less than or equal to 0.5, the jellyfish will move in a jellyfish swarm [27]. To determine the type of jellyfish motion inside a jellyfish swarm (passive motion and active motion), the function $1 - c(t)$ is used.

```
BEGIN
    SET the objective function (f(X) | X = x1, x2, ..., xn)                                  Problem definition
    SET bounds of search spaces, the maximum number of iterations (Max_iter), and population size (n_pop)   Controlling parameters definition
    INITIALIZE population of jellyfish (X_n | n = 1, 2, ..., n_pop) using a logistic map     Population initialization
    CALCULATE the quantity of food at each location of X_n (f(X_n))
    RECORD jellyfish with the best location (X*)
    SET iteration number: t = 1
    REPEAT
        FOR i = 1 to n_pop
            CALCULATE control time (c(t)) using Eq. (11)
            IF c(t) ≥ 0.5 THEN
                CALCULATE the new location of a jellyfish after it moves (X_i(t+1)) using Eq. (8)   Movement following ocean current
            ELSE
                IF rand(0,1) > 1-c(t) THEN                                                   Motions inside jellyfish swarm
                    CALCULATE the new location of a jellyfish after it moves (X_i(t+1)) using Eq. (9)   Passive motion
                ELSE
                    CALCULATE the new location of a jellyfish after it moves (X_i(t+1)) using Eq. (10)   Active motion
                END IF
            END IF
            CHECK boundary location
            CALCULATE the quantity of food at the new location of X_i(t+1)
            IF f(X_i(t+1)) > f(X_i(t)) THEN
                UPDATE the position of jellyfish
                and RECORD jellyfish with the best location (X*)
            END IF
        END FOR
        t = t+1
    UNTIL t > Max_iter
    PRINT the best result                                                                    Solution
END
```

Figure 19: Pseudocode of the jellyfish search algorithm.

When $\text{rand}(0,1)$ exceeds $(1 - c(t))$, the jellyfish will exhibit passive motion (type A). When $\text{rand}(0,1)$ is less than $(1 - c(t))$, the jellyfish will exhibit active motion (type B). As t increases, the value of $1 - c(t)$ also increases [27].

3.2.4. Algorithmic Flowchart and Pseudocode. The algorithmic flowchart and the pseudocode of the JS algorithm, starting from problem definition, controlling parameters' definition, initialization, to the loop of phases, are presented in Figures 18 and 19, respectively.

3.3. Validation and Performance Evaluation. Validating the capability of the DL model that classifies data or analyzes datasets to predict a new dataset is essential. In neural network models, a loss function usually refers to the minimization of the prediction error. The training error, which is the average loss of the training sample, is not useful for evaluating the performance of the model because a low training error may indicate that the model is overfitting the training data, and so will generally perform poorly given new data [48]. The validation, therefore, should be conducted using a separate sample error.

During the development of a DL model, a dataset is typically split into three sets–the training set, the validation set, and the test set. The training set is used to learn the pattern of the inputs that correspond to a certain output; the validation set is used to evaluate the prediction error of the training model and to tune its hyperparameters; the test set is used to assess the error of the final model. No exact rule for splitting the dataset exists, as the split depends on the number and complexity of the available data.

3.3.1. Validation Method. A validation set is used as the input of a previously trained prediction model to evaluate the performance of the model when used with new, never-seen-before data. The validation process is repeated multiple times with various hyperparameter combinations, and thus the purpose of using a validation set is to assess the performance of the training model and to find the optimal hyperparameters.

Two of the most popular methods for evaluating the generalization ability of the prediction model are holdout method and cross-validation. The holdout method randomly splits the data into a training set, a validation set, and a test set. The cross-validation method partitions a dataset into several subsets, implements the learning process on all but one of those subsets, and evaluates the performance using the left-out subset in turn. The cross-validation method is particularly suitable for a small dataset to enhance model validity.

For practical use in the ready-mixed concrete plant, the model is built based on the accumulated historical data, and subsequently will be used for a new concrete dataset in the prediction of compressive strength. To fairly reflect the

prediction accuracy on-site, this study adapted the holdout method by training/validating the model with the whole historical dataset and testing it with newly collected concrete data in the upcoming year. By doing so, one would not overestimate the model performance in practice and could prevent information leakage from model training.

3.3.2. Performance Metrics. The performance metrics that are used in this study are the mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), training time, and synthesis index (SI). The MAE is the average of the absolute differences between the actual and predicted values. Taking the absolute difference makes all error values positive, avoiding the false determination of an accurate prediction when negative and positive differences are summed.

Mean squared error (MSE) is the average of the squared differences between the actual and predicted values. The square root of the MSE, called the RMSE, is taken to the lower order of the MSE. MAPE is the average of the absolute errors divided by the actual values. The training time of various modeling techniques is compared to examine the implementation practicability.

A low value of MAE, RMSE, or MAPE indicates good performance; a short training time is desirable. SI is the mean of the sum of normalized values and indicates the overall model performance; it ranges from zero to one and zero indicates the best performance among all models. Table 1 provides the formulas for the performance measures.

Table 1: Performance metrics.

Performance metric	Formula
Mean absolute error (MAE)	$1/n\sum_{i=1}^{n}\lvert y - y'\rvert$
Mean squared error (MSE)	$1/n\sum_{i=1}^{n}(y' - y)^2$
Root mean squared error (RMSE)	$(1/n\sum_{i=1}^{n}(y' - y)^2)^{1/2}$
Mean absolute percentage error (MAPE)	$1/n\sum_{i=1}^{n}\lvert(y - y')/y\rvert$
Synthesis index (SI)	$1/m\sum_{i=1}^{m}\lvert(P - P_{\min})/(P_{\max} - P_{\min})\rvert$

Note. n, number of predictions; y, actual value; y', predicted value; m, number of performance metrics; P, value of the performance metric; $P_{\min}$, minimum value of performance metric; $P_{\max}$, maximum value of performance metric.

4. Analytical Results and Discussion

4.1. Experimental Settings

4.1.1. Software and Hardware. Model building and testing were implemented in Anaconda software with the Python programming language on a machine (computer) with an NVIDIA GeForce RTX 2080 Ti graphics card. The Jupyter Notebook application in Anaconda [49] was used to display the inputs and outputs of the prediction models. Python packages support specific programming tasks and protect against their incompatibility. Numerous Python packages, which are available for use with Anaconda, are used (such as NumPy, pandas, and matplotlib). For building and testing the DNN models, the TensorFlow package [50] is used. For building and testing the CNN-based models, the Keras Application package [51] is used.

In particular, the Keras Application package supports the implementation of CNN models for prediction, fine-tuning, and feature extraction. It provides CNN models with pretrained weights from "ImageNet." The package also provides a transfer learning feature that helps solve the practical problem of a lack of data resources and improves the accuracy of prediction using pretrained weights. Table 2 presents information about the models, with accuracies that were obtained using the 2012 ILSVRC ImageNet validation set [51]. The depth refers to the number of layers in the Keras Applications' CNN model, including the activation layer, batch normalization layer, and other layers.

4.1.2. Collection and Preprocessing of Data. A total of 8,310 data samples about ready-mixed concrete, relating to 32 variables, were collected from 2001 to 2019 by the Taiwan Construction Research Institute (TCRI). The data were split at the time of data sample collection to enable a prediction model to be built using historical data and tested using new data.

Accordingly, 339 data samples that covered one year (2019) were used in the testing process and the remaining 7,971 data samples were used in the training process. Of the 339 data samples for testing, 15 were removed because the value of compressive strength was missing, creating a test set of 324 data samples. The 7,971 data samples for training were further preprocessed according to the practical recommendations by a panel of domain experts in TCRI.

Among the 32 variables, the manufacturer's name, category of data, and date of collection were removed because the corresponding data were apparently irrelevant to the variability of concrete compressive strength. Ten other variables were removed because data were incomplete; these were the amount of admixture, the surface moisture content of sand (from a computer report and sieve analysis, respectively), silt charge, fineness modulus of sand, the strength of cement, specific surface area of cement, percentage of active blast furnace slag, fineness of blast furnace slag, and the ratio of water-reducing admixtures.

Totally, there are 19 concrete variables to be used for the prediction of the concrete compressive strength. One output variable is the test value of ready-mixed concrete compressive strength, and the other 18 input variables are the design strength of concrete, target strength of concrete, slump test, chloride ion content, temperature (temperature of the concrete taken on site), water-binder ratio, the water content of concrete, cementitious material consumption, cement ratio, amount of cement, amount of slag powder, amount of fly ash, amount of fine aggregate, amount of coarse aggregate, sand ratio, location (north), location (middle), and location (south).

The preprocessed data were processed again to yield three sets of data with different variables for use in numerical

Table 2: Convolutional neural network-based models in the Keras Application.

Model	Top 1 accuracy	Top 5 accuracy	Depth	Size (MB)	Parameters	Reference
VGG16	0.713	0.901	23	528	138,357,544	[33]
VGG19	0.713	0.900	26	549	143,667,240	
ResNet50	0.749	0.921	—	98	25,636,712	
ResNet101	0.764	0.928	—	171	44,707,176	[34]
ResNet152	0.766	0.931	—	232	60,419,944	
ResNet50V2	0.760	0.930	—	98	25,613,800	
ResNet101V2	0.772	0.938	—	171	44,675,560	[35]
ResNet152V2	0.780	0.942	—	232	60,380,648	
InceptionV3	0.779	0.937	159	92	23,851,784	[36]
InceptionResNetV2	0.803	0.953	572	215	55,873,736	[37]
Xception	0.790	0.945	126	88	22,910,480	[38]
MobileNet	0.704	0.895	88	16	4,253,864	[39]
MobileNetV2	0.713	0.901	88	14	3,538,984	[40]
DenseNet121	0.750	0.923	121	33	8,062,504	
DenseNet169	0.762	0.932	169	57	14,307,880	[41]
DenseNet201	0.773	0.936	201	80	20,242,984	
NASNetMobile	0.744	0.919	—	23	5,326,716	[42]
NASNetLarge	0.825	0.960	—	343	88,949,818	
EfficientNetB0	—	—	—	29	5,330,571	
EfficientNetB1	—	—	—	31	7,856,239	
EfficientNetB2	—	—	—	36	9,177,569	
EfficientNetB3	—	—	—	48	12,320,535	[43]
EfficientNetB4	—	—	—	75	19,466,823	
EfficientNetB5	—	—	—	118	30,562,527	
EfficientNetB6	—	—	—	166	43,265,143	
EfficientNetB7	—	—	—	256	66,658,687	

Table 3: Variables in the datasets.

Dataset variable	Dataset 1	Dataset 2	Dataset 3
Design strength of concrete	—	—	✓
Target strength of concrete	—	—	✓
Slump test	—	—	✓
Chloride ion content	—	—	✓
Temperature	—	—	✓
Water-binder ratio	✓	✓	✓
Water content of concrete	✓	✓	✓
Cementitious material consumption	✓	—	✓
Cement ratio	✓	—	✓
Amount of cement	✓	✓	✓
Amount of slag powder	✓	✓	✓
Amount of fly ash	✓	✓	✓
Amount of fine aggregate	✓	✓	✓
Amount of coarse aggregate	✓	✓	✓
Sand ratio	✓	—	✓
Location (north)	✓	—	✓
Location (middle)	✓	—	✓
Location (south)	✓	—	✓
Compressive strength test	✓	✓	✓

Note. Dataset 1 = industry recommendation; dataset 2 = suggested by research community; dataset 3 = all features considered. Variables in dataset 2 are frequently used to determine the compressive strength of concrete in the literature.

Table 4: Number of data points in the datasets.

Number of data points	Dataset 1	Dataset 2	Dataset 3
Number of total samples	6705	6705	5856
Number of training samples	6381	6381	5532
Number of testing samples	324	324	324
Number of input variables	13	7	18
Number of output variables	1	1	1

experiments for various purposes. Dataset 1 included 13 variables that are recommended by the TCRI; dataset 2 included 7 variables that are frequently used in prior studies [52–55] on the prediction of compressive strength; and dataset 3 included the resulting 18 variables after preprocessing. Tables 3–5 display the variables in the dataset, the number of data points in the datasets, and the descriptive statistics of variables in the datasets, respectively.

4.1.3. Converting Numerical Data into Images. The original numerical data were converted to images to be used as inputs to the CNN-based models. Each collection of values in a data sample for concrete was represented as an image. To create the image, the numerical data were first normalized to values between 0 and 1. These normalized data were then multiplied by 255 to encode them as grayscale values between 0 and 255 (Figure 20). Black represents 0 and white represents 255.

For each of datasets 1 and 2, a total of 6705 images were created. For dataset 3, a total of 5856 images were created. Figure 21 presents the example (dataset 3) of the labeling of the image data. Each image is labeled with the corresponding continuous output value, the compressive strength value of the ready-mixed concrete.

4.2. Implementation and Comparison. Prediction models and sensitivity experiments with various purposes were carried out (Table 6). Baseline models were used with the

Table 5: Descriptive statistics of variables from the datasets.

	Variables	Unit	Minimum	Maximum	Average
	Dataset 1—industry recommendation				
*X*6	Water-binder ratio	—	0.25	0.87	0.52
*X*7	Water content of concrete	(kg/m^3)	121.00	250.25	184.98
*X*8	Cementitious material consumption	(kg/m^3)	209.00	690.00	361.15
*X*9	Cement ratio	(%)	30.67	100.00	70.33
*X*10	Amount of cement	(kg/m^3)	99.20	507.00	255.43
*X*11	Amount of slag powder	(kg/m^3)	0.00	209.35	68.90
*X*12	Amount of fly ash	(kg/m^3)	0.00	180.00	36.82
*X*13	Amount of coarse aggregate	(kg/m^3)	344.24	1281.00	919.30
*X*14	Amount of fine aggregate	(kg/m^3)	468.00	1376.96	860.22
*X*15	Sand ratio	(%)	0.00	80.00	48.32
*X*16	Location (north)	—	0.00	1.00	0.44
*X*17	Location (middle)	—	0.00	1.00	0.12
*X*18	Location (south)	—	0.00	1.00	0.44
Y	Compressive strength test	(kgf/cm^2)	125.00	724.00	343.49
	Dataset 2—suggested by the research community				
*X*6	Water-binder ratio	—	0.25	0.87	0.52
*X*7	Water content of concrete	(kg/m^3)	121.00	250.25	184.98
*X*10	Amount of cement	(kg/m^3)	99.20	507.00	255.43
*X*11	Amount of slag powder	(kg/m^3)	0.00	209.35	68.90
*X*12	Amount of fly ash	(kg/m^3)	0.00	180.00	36.82
*X*13	Amount of fine aggregate	(kg/m^3)	468.00	1376.96	860.22
*X*14	Amount of coarse aggregate	(kg/m^3)	344.24	1281.00	919.30
Y	Compressive strength test	(kgf/cm^2)	125.00	724.00	343.49
	Dataset 3—all features considered				
*X*1	Design strength of concrete	(kgf/cm^2)	140.00	420.00	254.40
*X*2	Target strength of concrete	(kgf/cm^2)	160.00	660.00	320.27
*X*3	Slump test	(cm)	8.50	69.00	19.48
*X*4	Chloride ion content	(%)	0.00	0.14	0.04
*X*5	Temperature	(°C)	14.00	35.00	26.19
*X*6	Water-binder ratio	—	0.25	0.83	0.52
*X*7	Water content of concrete	(kg/m^3)	121.00	250.25	184.87
*X*8	Cementitious material consumption	(kg/m^3)	209.00	690.00	363.42
*X*9	Cement ratio	(%)	30.67	100.00	70.10
*X*10	Amount of cement	(kg/m^3)	99.20	507.00	256.17
*X*11	Amount of slag powder	(kg/m^3)	0.00	209.35	68.85
*X*12	Amount of fly ash	(kg/m^3)	0.00	180.00	38.40
*X*13	Amount of fine aggregate	(kg/m^3)	468.00	1376.96	860.88
*X*14	Amount of coarse aggregate	(kg/m^3)	344.24	1281.00	916.43
*X*15	Sand ratio	(%)	0.00	80.00	48.41
*X*16	Location (north)	—	0.00	1.00	0.48
*X*17	Location (middle)	—	0.00	1.00	0.13
*X*18	Location (south)	-	0.00	1.00	0.39
Y	Compressive strength	(kgf/cm^2)	162.00	724.00	344.82

hyperparameters set to default values in the TensorFlow and Keras Applications. In the DNN, numerical data are input, while for the CNN-based models, the input numerical data are converted to image data. In this study, the size of the image input to each CNN-based model was the minimum possible size to meet the practical needs.

4.2.1. Deep Learning Models and Performance. Since the same model and hyperparameters yielded different model performance values in different runs, each model was tested five times and the average model performance value was taken as the actual. For both the CNN and DNN models, the loss function was set to be the MSE. In the DNN model, 50 hidden layers with selected numbers of hidden nodes (Table 7) had the best prediction accuracy in comparison with other numbers of hidden layers and other numbers of hidden nodes. The architecture was thus used to build the baseline DNN prediction model.

Tables 8–10 compare the performances of the DL models in predicting the compressive strength of ready-mixed concrete when they are trained and tested using the given data. The results indicate that the CNN models, ResNet50V2, MobileNet, and DenseNet121, with their default parameters, all performed best on the three datasets, respectively. The CNN models, ResNet50V2, MobileNet, and DenseNet121, with image data, outperformed the baseline DNN model with numerical data. The results also indicate that the best CNN models on each dataset outperformed the DNN in terms of each performance metric, except for the training time.

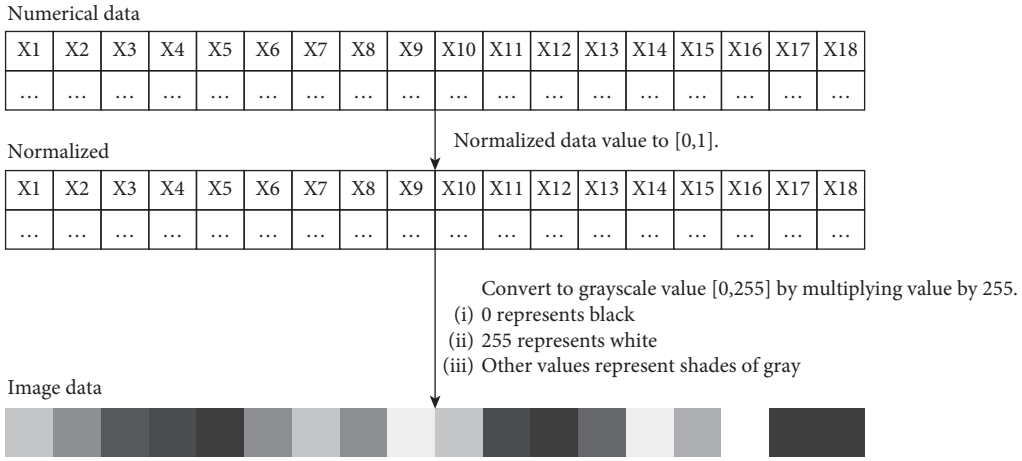


Figure 20: Conversion example of numerical data to image data.

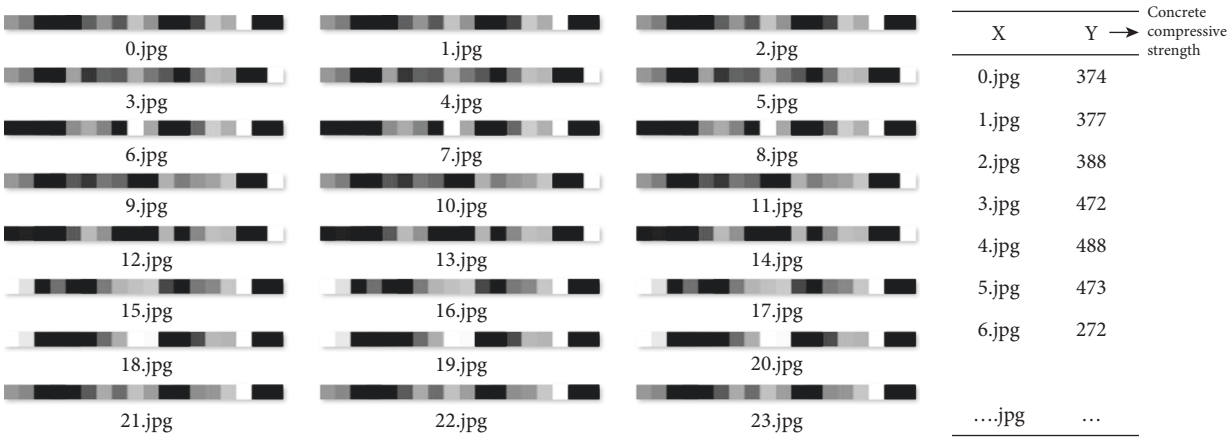


Figure 21: Input images and corresponding output labels.

Table 6: Experimental settings.

Research task	Data type	Purpose	Method
Comparison of deep learning models	Numerical data and image data	Search for the best CNN model (using image data) and compare the best CNN model with a DNN model (using numerical data)	CNNs and DNN: VGG, ResNet, ResNetV2, InceptionV3, InceptionResNetV2, MobileNets, MobileNetV2, NASNet, EfficientNets, DenseNet
Construction of optimized deep learning models	Image data	Enhance the best-performing models with optimized hyperparameters	Optimizing deep learning models by jellyfish search algorithm

Table 7: Number of hidden nodes in each hidden layer of the deep neural network (DNN).

Hidden layer	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}–10^{th}	11^{th}–20^{th}	21^{st}–30^{th}	31^{st}–40^{th}	41^{st}–50^{th}
Number of hidden nodes	4096	2048	1024	512	256	128	64	32	16	8	4	2

4.2.2. Optimized Convolutional Neural Network-Based Models. As CNN models, ResNet50V2, MobileNet, and DenseNet121, performed best in the corresponding datasets, a metaheuristic optimization algorithm, the jellyfish search (JS) optimizer, was used to optimize them. The CNN models were optimized to minimize the errors of prediction of the compressive strength of ready-mixed concrete using the best values of the hyperparameters.

The JS optimizer was used to find the best hyperparameter values in a set of ranges. Several hyperparameters

Table 8: Deep learning model performance on dataset 1.

Model	Training time (h:m:s)	MAPE (%)	RMSE (kgf/cm^2)	MAE (kgf/cm^2)	SI
Xception	2:31:01	14.0264	76.4217	57.7471	0.285
VGG16	0:17:58	15.9598	85.6755	65.6588	0.249
VGG19	0:21:16	14.8719	79.9609	61.1835	0.147
ResNet50	0:18:28	15.0252	80.8499	62.5910	0.164
ResNet101	0:32:30	14.3817	75.4145	57.5325	0.091 (3)
ResNet152	0:44:12	14.3462	78.5345	59.2920	0.144
ResNet50V2	0:16:56	13.8000	73.7818	56.4419	0.027 (1)
ResNet101V2	0:29:17	14.7393	74.4318	58.7149	0.100
ResNet152V2	0:43:11	16.2188	85.7025	67.9747	0.318
InceptionV3	0:45:14	14.2727	75.2054	58.0613	0.111
InceptionResNetV2	1:43:38	14.7849	77.9702	60.6185	0.264
MobileNet	0:10:35	15.1504	79.4783	62.0158	0.141
MobileNetV2	0:12:09	17.2442	82.0049	63.9462	0.244
DenseNet121	0:18:49	15.6411	82.3700	64.8698	0.213
DenseNet169	0:24:41	15.3712	80.3540	63.4890	0.190
DenseNet201	0:31:28	15.4292	80.7504	63.7175	0.207
NASNetMobile	0:35:50	15.8436	82.1353	64.5839	0.244
EfficientnetB0	0:18:10	14.2585	75.3426	58.3330	0.069 (2)
EfficientnetB1	0:27:30	14.9977	80.3189	61.8780	0.169
EfficientnetB2	0:28:38	15.4503	80.6015	63.4244	0.200
EfficientnetB3	0:35:30	14.9118	80.7490	61.6706	0.181
EfficientnetB4	0:45:16	14.6289	79.1150	61.1313	0.173
EfficientnetB5	0:59:30	14.5750	76.2138	59.6796	0.164
EfficientnetB6	1:13:08	15.0979	81.5700	62.4580	0.261
EfficientnetB7	1:42:25	14.4783	74.6023	59.1117	0.218
DNN	0:00:46	21.4910	112.2759	87.9198	0.750

Table 9: Deep learning model performance on dataset 2.

Model	Training time (h:m:s)	MAPE (%)	RMSE (kgf/cm^2)	MAE (kgf/cm^2)	SI
Xception	1:12:19	17.2430	89.4229	70.1275	0.493
VGG16	0:10:26	18.3202	100.0902	73.6012	0.387
VGG19	0:12:32	17.7289	98.4250	74.1325	0.373
ResNet50	0:12:24	15.0208	78.1490	61.1607	0.105
ResNet101	0:21:34	14.5379	74.5514	58.2454	0.086 (2)
ResNet152	0:30:17	16.5355	90.8698	67.9069	0.321
ResNet50V2	0:11:28	15.7710	80.7428	63.8803	0.154
ResNet101V2	0:20:19	15.1000	76.9871	60.4406	0.124
ResNet152V2	0:28:43	14.4184	73.6713	57.4732	0.098
InceptionV3	0:27:04	16.4996	87.7770	69.3648	0.302
InceptionResNetV2	1:02:49	15.8074	84.5667	66.3267	0.371
MobileNet	0:06:18	15.0699	77.6579	61.3514	0.084 (1)
MobileNetV2	0:07:26	18.6955	83.8692	68.5122	0.264
DenseNet121	0:12:41	15.3784	83.6683	64.4323	0.167
DenseNet169	0:17:17	15.5968	85.7416	65.6683	0.209
DenseNet201	0:22:02	14.9204	83.5047	62.5952	0.175
NASNetMobile	0:20:32	22.3714	115.6740	93.5611	0.745
EfficientnetB0	0:12:27	15.0674	80.2897	61.9535	0.124
EfficientnetB1	0:17:52	15.5417	81.8874	63.5534	0.174
EfficientnetB2	0:18:12	14.5659	77.0040	59.4023	0.096 (3)
EfficientnetB3	0:21:51	15.4034	81.4909	63.2014	0.180
EfficientnetB4	0:27:17	15.7882	83.7080	64.3034	0.228
EfficientnetB5	0:36:30	14.9359	78.5506	60.4780	0.185
EfficientnetB6	0:45:47	15.0286	78.3096	61.4788	0.225
EfficientnetB7	1:01:55	15.3541	80.7350	62.9946	0.313
DNN	0:00:40	23.9215	119.4923	95.5619	0.750

Table 10: Deep learning model performance on dataset 3.

Model	Training time (h:m:s)	MAPE (%)	RMSE (kgf/cm^2)	MAE (kgf/cm^2)	SI
Xception	2:59:21	13.0524	64.3952	51.9907	0.326
VGG16	0:20:24	13.8902	69.6804	56.1444	0.165
VGG19	0:23:28	14.1965	69.1497	55.3852	0.169
ResNet50	0:20:26	13.9585	70.9084	57.2012	0.178
ResNet101	0:34:28	11.6479	59.5678	47.1273	0.049 (2)
ResNet152	0:48:51	11.7348	60.4651	47.2528	0.076
ResNet50V2	0:18:31	12.3769	64.2470	51.1966	0.083
ResNet101V2	0:31:55	12.1501	63.4072	49.7640	0.086
ResNet152V2	0:46:54	12.0281	61.4381	48.0813	0.087
InceptionV3	0:51:37	13.3665	64.9479	52.1857	0.157
InceptionResNetV2	1:57:42	13.5490	64.0620	51.9634	0.248
MobileNet	0:12:27	13.1265	65.2462	52.4753	0.100
MobileNetV2	0:14:00	13.0155	60.0301	47.8787	0.053
DenseNet121	0:20:09	11.7167	59.3511	47.1034	0.029 (1)
DenseNet169	0:25:51	11.7929	61.0411	47.9452	0.051 (3)
DenseNet201	0:32:50	11.6047	60.6139	47.7109	0.054
NASNetMobile	0:41:08	24.6483	109.9440	93.7760	0.780
EfficientnetB0	0:20:37	12.9179	64.7033	52.1055	0.103
EfficientnetB1	0:30:28	13.3862	68.6528	55.0783	0.159
EfficientnetB2	0:31:25	13.4363	68.1339	54.9089	0.159
EfficientnetB3	0:38:59	13.2016	67.0390	53.1422	0.150
EfficientnetB4	0:49:39	13.1075	65.5639	52.3979	0.153
EfficientnetB5	1:09:29	13.2787	67.5948	54.1713	0.203
EfficientnetB6	1:26:14	12.4606	63.7319	50.5952	0.174
EfficientnetB7	1:55:15	12.8867	63.4471	51.0061	0.224
DNN	0:00:44	22.8024	116.0370	90.6988	0.698

Table 11: Hyperparameter settings for deep learning models.

Hyperparameter	Literature value	Search range in this study
	ResNet50V2 [34]	
Batch normalization-epsilon	$1.001e-5$	$[1.001e-5, 0.00005, 0.0001, 0.0005, 0.001]$
Batch size	64, 256	[8, 16, 32, 64]
Epochs	40, 90, 300	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
ADAM-learning rate	0.1	[0.001, 0.005, 0.01, 0.05, 0.1]
Dropout rate	0.5	0.00–0.99
	MobileNet [39]	
Batch size	64, 256	[8, 16, 32, 64]
Epochs	40, 90, 300	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
ADAM-learning rate	0.1	[0.001, 0.005, 0.01, 0.05, 0.1]
Dropout rate	0.5	0.00–0.99
	DenseNet121 [41]	
Growth rate	32	12–48
Batch normalization-epsilon	$1.001e-5$	$[1.001e-5, 0.00005, 0.0001, 0.0005, 0.001]$
Batch size	64, 256	[8, 16, 32, 64]
Epochs	40, 90, 300	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
Reduction	0.5	0.1–1.0
ADAM-learning rate	0.1	[0.001, 0.005, 0.01, 0.05, 0.1]
Dropout rate	0.2	0.00–0.99

Table 12: Performance of the best and optimized CNN models.

Dataset	Model	MAPE (%)	RMSE (kgf/cm^2)	MAE (kgf/cm^2)
1	ResNet50V2	13.8000	73.7818	56.4419
	JS-ResNet50V2	13.1327	68.5794	52.4591
2	MobileNet	17.0406	91.6945	71.1198
	JS-MobileNet	17.0671	90.4711	70.0870
3	DenseNet121	11.7167	59.3511	47.1034
	JS-DenseNet121	11.5443	58.4346	45.8917

Table 13: Optimal hyperparameters of the best CNN models.

Hyperparameter	Optimal value
JS-ResNet50V2	
Batch normalization-epsilon	0.0005
Batch size	64
Epochs	100
ADAM-learning rate	0.001
Dropout rate	0.26
JS-MobileNet	
Batch size	16
Epochs	70
ADAM-learning rate	0.001
Dropout rate	0.65
JS-DenseNet121	
Growth rate	38
Batch normalization-epsilon	0.00005
Batch size	64
Epochs	90
Reduction	0.7
ADAM-learning rate	0.001
Dropout rate	0.33

Table 14: Sensitivity analysis of input features.

No.	*X*1	*X*2	*X*3	*X*4	*X*5	*X*6	*X*7	*X*8	*X*9	*X*10	*X*11	*X*12	*X*13	*X*14	*X*15	*X*16	*X*17	*X*18	MAPE (%)
1	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.94
2	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.91
3	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.95
4	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.66
5	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.99
6	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.01
7	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.32
8	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.05
9	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.57
10	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	✓	11.17
11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	✓	11.63
12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓	11.69
13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	11.29
14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓	11.59
15	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	11.26
16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	11.98
17	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	11.72

Note. *X*1 = design strength of concrete, *X*2 = target strength of concrete, *X*3 = slump test, *X*4 = chloride ion content, *X*5 = temperature, *X*6 = water-binder ratio, *X*7 = water content of concrete, *X*8 = cementitious material consumption, *X*9 = cement ratio, *X*10 = amount of cement, *X*11 = amount of slag powder, *X*12 = amount of fly ash, *X*13 = amount of fine aggregate, *X*14 = amount of coarse aggregate, *X*15 = sand ratio, *X*16 = location (north), *X*17 = location (middle), and *X*18 = location (south).

of a CNN, such as the epsilon of batch normalization, batch size, epoch, learning rate, and dropout rate, were selected to be adjusted during the search herein. For DenseNet121, two additional hyperparameters were optimized—the growth rate and the reduction value. Table 11 presents the default values of hyperparameters in the reference papers [34, 41] and the range of hyperparameters to be finetuned in this study.

Table 12 compares the performances of best CNN models using default hyperparameters and optimized by JS in predicting the compressive strength of ready-mixed concrete. The results indicate that using the JS optimizer on the hyperparameters improved the accuracy of the prediction models. Table 13 shows the best hyperparameter settings for each optimized CNN model.

4.3. Influence of Feature and Image Pixel Orientation on Modeling Accuracy. To examine the sensitivity of the generalization ability of a prediction model, the resulting 18 variables (features) were experimented using the best CNN model (DenseNet121) by removing one of the variables and using the remaining variables for model training. For the location variables, three variables were removed

Table 15: Correlation between the feature and compressive strength of ready-mixed concrete.

Feature	Correlation coefficient between feature and Y
$X1$	0.75
$X2$	0.82
$X3$	0.23
$X4$	0.05
$X5$	−0.15
$X6$	−0.74
$X7$	−0.15
$X8$	0.73
$X9$	0.14
$X10$	0.46
$X11$	0.05
$X12$	0.02
$X13$	−0.44
$X14$	−0.06
$X15$	−0.25
$X16$	0.24
$X17$	0.06
$X18$	−0.29

Note. $X1$ = design strength of concrete, $X2$ = target strength of concrete, $X3$ = slump test, $X4$ = chloride ion content, $X5$ = temperature, $X6$ = water-binder ratio, $X7$ = water content of concrete, $X8$ = cementitious material consumption, $X9$ = cement ratio, $X10$ = amount of cement, $X11$ = amount of slag powder, $X12$ = amount of fly ash, $X13$ = amount of fine aggregate, $X14$ = amount of coarse aggregate, $X15$ = sand ratio, $X16$ = location (north), $X17$ = location (middle), $X18$ = location (south), and Y = compressive strength of ready-mixed concrete.

Table 16: Results of the order importance analysis of the image-like dataset.

Image pixel orientation	Type of pixel order	MAPE (%)	RMSE (kgf/cm^2)	MAE (kgf/cm^2)
Original order	Random arrangement	11.5443	58.4346	45.8917
Correlated order	Descending by correlated values	12.0831	61.3435	48.1922
	Descending by absolute correlated values	12.5888	64.5037	50.7435

simultaneously and the remaining variables were used for sensitivity analysis. These tests were conducted to investigate the effect of each feature (attribute) on the generalization ability of model prediction. Table 14 displays the performance results with MAPEs, in which the lower value of the MAPE stands for the better model performance without the specified attribute. The experiment demonstrated that the MAPEs do not differ much from one another. However, the slight increase of MAPE in each numerical experiment comparing to the baseline MAPE (11.72%) implies the inclusion of those variables ($X1$–$X3$, $X5$, and $X16$–$X18$) has a positive impact on the prediction accuracy of ready-mixed concrete compressive strength.

Another numerical experiment was conducted to examine the influence of image pixel orientation (pixel row order) on the computer vision-based modeling performance. Two types of image pixel orientation (IPO) formed by the input attributes (pixels) were tested, namely, the original pixel array and the correlated pixel array, according to the correlation values between the input attributes and the compressive strength. Specifically, the input image data were shaped by arranging the input attributes (pixels) in random order and descending the pixels order by their correlation coefficients, respectively.

Table 15 displays the correlation coefficients between the input variables and the compressive strength of ready-mixed concrete. Ordering the IPO based on the magnitude of the correlation coefficients, two new datasets were created. One IPO is arranged by descending the original values of the correlation coefficients and the other IPO is arranged by descending their absolute values.

Table 16 presents the sensitivity analysis of image pixel orientation on the computer vision-based modeling performance. It is observable that all metrics with the correlated order of image pixel orientation show worse performance than that obtained using the original ordered image by the same optimized CNN model (JS-DenseNet121). Therefore, the analytical results indicate that the correlated order of image pixel orientation for the image converting of ready-mixed concrete data does not significantly influence the performance of the prediction model.

5. Conclusions

The effectiveness of computer vision in predicting the compressive strength of ready-mixed concrete was analyzed to improve the predictions thereof. Deep learning (DL) models were constructed by imaging the numerical data as inputs to predict the compressive strength of ready-mixed concrete. Various prediction models were compared and the best DL prediction models were identified for different sets of input concrete-related features and optimized after their performances were further analyzed.

The models for the prediction of concrete compressive strength are frequently built with the use of cross-validation or random split in-sample data for evaluating prediction

accuracy, which often gives optimistic results (overfitting) in the training/test process while exhibiting poor performance in future use. It's mainly because the processes, materials, machines, and technicians that are involved to manufacture ready-mixed concrete in batch plants are being continually improved and replaced periodically. Up-to-date samples for ready-mixed concrete might be derived differently from the evolving development of batch processes.

A prediction model is built using historical data; it uses newly collected data, which should be irrelevant to the training data samples, to make predictions; therefore, the optimality of using random split in-sample data to test models in the prediction of concrete compressive strength in the literature is now doubted. To capture the actual performance of predicting the compressive strength of concrete, out-of-sample data (newly collected data) should be used for model testing to avoid potential information leakage. Although the model accuracy may be decreased in comparison with that obtained by in-sample cross-validation or randomly split data for training and test, using such an approach for the out-of-sample test reflects the real predictive performance in practice.

Furthermore, CNN-oriented models are often trained without tuning the hyperparameters. This study adopts a metaheuristic optimization algorithm to optimize the prediction model. The predictive accuracy of computer vision-based deep learning models was improved herein using the jellyfish search (JS) metaheuristic optimization algorithm. The JS optimizer finds the best hyperparameters, optimizing the performance metrics of the CNN models. This study contributes to the novel application of the computer vision-based method, which integrates the latest CNN models with a newly developed JS optimizer to predict the compressive strength of ready-mixed concrete. The analytical experiments show that modeling with image-converting data outperforms the models using the original numerical data.

In this investigation, the training data were samples on ready-mixed concrete only. Using data on high-performance concrete or more complex engineering data would improve this work of the computer vision approach to predicting a numerical output like the compressive strength of concrete. More cases should be studied to confirm the effectiveness of imaging data on ready-mixed concrete and other types of concrete to identify patterns of compressive strength by the bio-inspired metaheuristic optimization of computer vision-based deep learning models.

Future studies could consider environment-oriented factors that may affect the ready-mixed concrete compressive strength, such as the type of manufacturing equipment, transporting process of concrete, and the handling speed of on-site operators in addition to the material-oriented attributes herein. A fair comparison between laboratory-determined concrete compressive strength and on-site evaluation of concrete compressive strength should be investigated.

Abbreviations

AI:	Artificial intelligence
ANN:	Artificial neural network
BN:	Batch normalization
CNN/ConvNet:	Convolutional neural network
DenseNet:	Dense convolutional network
DL:	Deep learning
DNN:	Deep neural network
FNN:	Feedforward neural network
IN:	Instance normalization
IPO:	Image pixel orientation
JS:	Jellyfish search
LN:	Layer normalization
MAE:	Mean absolute error
MAPE:	Mean absolute percentage error
MSE:	Mean squared error
NASNet:	Neural architecture search network
ReLU:	Rectified linear unit
ResNet:	Residual neural network
RF:	Random forest
RMSE:	Root mean squared error
RNN:	Recurrent neural network
SI:	Synthesis index
SVR:	Support vector regression
TCRI:	Taiwan Construction Research Institute
VGG:	Visual geometry group
Xception:	Extreme inception.

Symbols

$w \times h \times c$:	Width of image × height of image × number of channels
m:	Dimension of an input image
n:	Filter size
C:	Convolution map
I:	Input image data
Θ:	Convolution operation
F:	Filter
o:	Dimension of convolution map
s:	Stride
zp:	Zero padding
f:	Nonlinear activation function
C_m:	Convolution map after applying the nonlinear activation function f
P_m:	Pooling map
P_o:	Pooling operation
Y^i:	Model output of the ith fully connected hidden layer
H^i:	Weight sum vector
B^i:	The activation level of the artificial neurons
$BN(x)$:	Batch normalization at a given layer from x
γ:	Scale parameter for the channel
β:	Shift parameter for the channel
μ_B:	Mean of the batch

σ_B: Standard deviation of the batch
$*$: Element-wise product
$\underline{f}^l(x_i)$: Original feature
$\tilde{f}^l(x_i)$: Distorted features
m_i^l: Binary mask
$\underline{\mathrm{d}}^{\mathrm{l}}$: Dimension of the feature map of the l-th layer
$\overrightarrow{\text{trend}}$: Direction of the ocean current
X^*: Jellyfish with the optimal location
μ: Average location of all jellyfish
X_i: Jellyfish of interest
t: Time specified as an iteration number
$X_i(t)$: Current location of a jellyfish
$X_i(t+1)$: New location of a jellyfish after a movement
$\text{rand}(0,1)$: Random number between 0 and 1
Ub: Upper bounds of the search spaces
Lb: Lower bounds of the search spaces
X_j: Jellyfish other than the jellyfish of interest
$f(X_i)$: Quantity of food at the location of X_i
$f(X_j)$: Quantity of food at the location of X_j
$\overrightarrow{\text{Direction}}$: Direction of active motion (type B) of jellyfish
$c(t)$: Time control function
Max_{iter}: Maximum number of iterations
n: Number of predictions
y: Actual value
y': Predicted value
m: Number of performance metrics
P: Value of performance metric
$P_{\min}$: Minimum value of performance metric
$P_{\max}$: Maximum value of performance metric.

Data Availability

The datasets, codes, and replication of results generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Taiwan Construction Research Institute and the Ministry of Science and Technology, Taiwan, for financially supporting this research under grants NTUST-TCRI-No.109-0139-9257 and MOST 109-2221-E-011-040-MY3, respectively.

References

[1] Y. Kellouche, B. Boukhatem, M. Ghrici, and A. Tagnit-Hamou, "Exploring the major factors affecting fly-ash concrete carbonation using artificial neural network," *Neural Computing & Applications*, vol. 31, no. 6, pp. 969–988, 2019.

[2] M. Adil, R. Ullah, S. Noor, and N. Gohar, "Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design," *Neural Computing & Applications*, vol. 34, no. 11, pp. 8355–8363, 2022.

[3] O. R. Abuodeh, J. A. Abdalla, and R. A. Hawileh, "Assessment of compressive strength of Ultra-high Performance Concrete using deep machine learning techniques," *Applied Soft Computing*, vol. 95, Article ID 106552, 2020.

[4] F. Deng, Y. He, S. Zhou, Y. Yu, H. Cheng, and X. Wu, "Compressive strength prediction of recycled concrete based on deep learning," *Construction and Building Materials*, vol. 175, pp. 562–569, 2018.

[5] A. Sharma, B. Tirumuruhan, G. S. Muthuvel, A. K. Gupta, and R. Sujith, "Optimization of process parameters of boron carbide-reinforced Al-Zn-Mg-Cu matrix composite produced by pressure-assisted sintering," *Journal of Materials Engineering and Performance*, vol. 31, no. 1, pp. 328–340, 2022.

[6] M. Lessard, O. Chaalla, and P.-C. Aitcin, "Testing high-strength concrete compressive strength," *ACI Materials Journal*, vol. 90, no. 4, pp. 303–308, 1993.

[7] A. E. Ben-Zeitun, "Use of pulse velocity to predict compressive strength of concrete," *International Journal of Cement Composites and Lightweight Concrete*, vol. 8, no. 1, pp. 51–59, 1986.

[8] K. L. Chung, L. Wang, M. Ghannam, M. Guan, and J. Luo, "Prediction of concrete compressive strength based on early-age effective conductivity measurement," *Journal of Building Engineering*, vol. 35, Article ID 101998, 2021.

[9] M. F. M. Zain and S. M. Abd, "Multiple regression model for compressive strength prediction of high performance concrete," *Journal of Applied Sciences*, vol. 9, no. 1, pp. 155–160, 2009.

[10] B. H. Bharatkumar, R. Narayanan, B. K. Raghuprasad, and D. S. Ramachandramurthy, "Mix proportioning of high performance concrete," *Cement and Concrete Composites*, vol. 23, no. 1, pp. 71–80, 2001.

[11] J. V. R. Luke, M. Snell, and D. W. Norval, "Predicting early concrete strength," *Concrete International*, vol. 11, no. 12, 1989, https://www.concrete.org/publications/internationalconcreteabstractsportal/m/details/id/2092.

[12] Y. Essam, P. Kumar, A. N. Ahmed, M. A. Murti, and A. El-Shafie, "Exploring the reliability of different artificial intelligence techniques in predicting earthquake for Malaysia," *Soil Dynamics and Earthquake Engineering*, vol. 147, Article ID 106826, 2021.

[13] H. Jin and J. Zhao, "Real-time energy consumption detection simulation of network node in internet of things based on artificial intelligence," *Sustainable Energy Technologies and Assessments*, vol. 44, Article ID 101004, 2021.

[14] H. Adeli, *Four Decades of Computing in Civil Engineering*Springer Singapore, Singapore, 2020.

[15] Y. Xu, Y. Zhou, P. Sekula, and L. Ding, "Machine Learning in Construction: From Shallow to Deep Learning," *Developments in the Built Environment*, vol. 6, no. 6, Article ID 100045, 2021.

[16] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[17] B. Korbar, A. M. Olofson, A. P. Miraflor et al., "Deep learning for classification of colorectal polyps on whole-slide images," *Journal of Pathology Informatics*, vol. 8, no. 1, p. 30, 2017.

[18] J.-S. Chou, M. A. Karundeng, D.-N. Truong, and M.-Y. Cheng, "Identifying deflections of reinforced concrete beams under seismic loads by bio-inspired optimization of deep residual learning," *Structural Control and Health Monitoring*, vol. 29, no. 4, Article ID e2918, 2022.

[19] W. G. Hatcher and W. Yu, "A survey of deep learning: platforms, applications and emerging research trends," *IEEE Access*, vol. 6, pp. 24411–24432, 2018.

[20] N. Chen, S. B. Zhao, Z. W. Gao et al., "Virtual mix design: prediction of compressive strength of concrete with industrial wastes using deep data augmentation," *Construction and Building Materials*, vol. 323, p. 13, 2022.

[21] H. R. Maier, S. Razavi, Z. Kapelan, L. S. Matott, J. Kasprzyk, and B. A. Tolson, "Introductory overview: optimization using evolutionary algorithms and other metaheuristics," *Environmental Modelling & Software*, vol. 114, pp. 195–213, 2019.

[22] Q. Liu, X. Li, H. Liu, and Z. Guo, "Multi-objective metaheuristics for discrete optimization problems: a review of the state-of-the-art," *Applied Soft Computing*, vol. 93, Article ID 106382, 2020.

[23] P. E. Black, "Greedy Algorithm, Dictionary of Algorithms and Data Structures 2021," 2005, http://www.nist.gov/dads.

[24] P. Baldi, "Gradient descent learning algorithm overview: a general dynamical systems perspective," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 182–195, 1995.

[25] J. Pearl, "Heuristics: intelligent search strategies for computer problem solving," 1984, https://www.gbv.de/dms/weimar/toc/021186472_toc.pdf.

[26] J.-S. Chou and N.-M. Nguyen, "FBI inspired meta-optimization," *Applied Soft Computing*, vol. 93, Article ID 106339, 2020.

[27] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Applied Mathematics and Computation*, vol. 389, Article ID 125535, 2021.

[28] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[29] N. Wagaa, H. Kallel, and N. Mellouli, "Improved Arabic alphabet characters classification using convolutional neural networks (CNN)," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 9965426, 16 pages, 2022.

[30] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for Activation Functions," 2017, https://arxiv.org/abs/1710.05941.

[31] M. Koçyiğit, L. Sevilla-Lara, T. M. Hospedales, and H. Bilen, "Unsupervised batch normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 918-919, Seattle, WA, USA, June 2020.

[32] Y. Tang, Y. Wang, Y. Xu et al., "Beyond dropout: feature map distortion to regularize deep neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5964–5971, 2020.

[33] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, https://arxiv.org/abs/1409.1556.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *Computer Vision - ECCV 2016*, vol. 9908, pp. 630–645, 2016.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, February 2017.

[38] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[39] A. G. Howard, M. Zhu, B. Chen et al., "Efficient Convolutional Neural Networks for mobile Vision Applications," 2017, https://arxiv.org/abs/1704.04861.

[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.

[41] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.

[42] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, San Francisco, CA, USA, June 2018.

[43] M. Tan and Q. Le, "Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks," pp. 6105–6114, 2019, https://arxiv.org/abs/1905.11946.

[44] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

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

[46] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[47] R. Ma, M. Karimzadeh, A. Ghabussi et al., "Assessment of Composite Beam Performance Using GWO–ELM Metaheuristic Algorithm," *Engineering with Computers*, 2021.

[48] K. Nordhausen, "The elements of statistical learning: data mining, inference, and prediction, second edition by trevor hastie, robert tibshirani, jerome friedman," in *International Statistical Review*, T. Hastie, R. Tibshirani, and J. Friedman, Eds., vol. 773, p. 482, Second Edition, 2009.

[49] A. S. Distribution, "Conda, Anaconda," 2021, https://www.anaconda.com/products/distribution.

[50] M. Abadi, A. Agarwal, P. Barham et al., "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2016, https://arxiv.org/abs/1603.04467.

[51] F. Chollet, *Deep Learning with Python*, Manning Publications Co, Shelter Island, NY, USA, 2021.

[52] J.-S. Chou and A.-D. Pham, "Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength," *Construction and Building Materials*, vol. 49, pp. 554–563, 2013.

[53] A. Khashman and P. Akpinar, "Non-destructive prediction of concrete compressive strength using neural networks," *Procedia Computer Science*, vol. 108, pp. 2358–2362, 2017.

[54] Q. Han, C. Gui, J. Xu, and G. Lacidogna, "A generalized method to predict the compressive strength of high-performance concrete by improved random forest algorithm," *Construction and Building Materials*, vol. 226, pp. 734–742, 2019.

[55] H. Y. Zhang, X. W. Cheng, Y. Li, and X. L. Du, "Prediction of failure modes, strength, and deformation capacity of RC shear walls through machine learning," *Journal of Building Engineering*, vol. 50, p. 22, 2022.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 1279945, 14 pages
https://doi.org/10.1155/2022/1279945

Research Article

Leveraging a Neuroevolutionary Approach for Classifying Violent Behavior in Video

Carlos Flores-Munguía, José C. Ortiz-Bayliss, and Hugo Terashima-Marín

Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias Ave, Eugenio Garza Sada 2501 Sur Col, Tecnológico C.P. 64849, Monterrey, Nuevo Leon, Mexico

Correspondence should be addressed to José C. Ortiz-Bayliss; jcobayliss@tec.mx

Received 18 February 2022; Revised 22 June 2022; Accepted 4 July 2022; Published 15 July 2022

Academic Editor: Diego Oliva

Security has become a critical issue for complex and expensive systems and day-to-day situations. In this regard, the analysis of surveillance cameras is a critical issue usually restricted to the number of people devoted to such a task, their knowledge and judgment. Nonetheless, different approaches have arisen to automate this task in recent years. These approaches are mainly based on machine learning and benefit from developing neural networks capable of extracting underlying information from input videos. Despite how competent those networks have proved to be, developers must face the challenging task of defining both the architecture and hyperparameters that allow such networks to work adequately and optimize the use of computational resources. In short, this work proposes a model that generates, through a genetic algorithm, neural networks for behavior classification within videos. Two types of neural networks evolved as part of this work, shallow and deep, which are structured on dense and 3D convolutional layers. Each network requires a particular type of input data: the evolution of the pose of people in the video and video sequences, respectively. Shallow neural networks use a direct encoding approach to map each part of the chromosome into a phenotype. In contrast, deep neural networks use indirect encoding, blueprints representing entire networks, and modules to depict layers and their connections. Our approach obtained relevant results when tested on the Kranok-NV dataset and evaluated with standard metrics used for similar classification tasks.

1. Introduction

Nowadays, violence harms people physically and mentally. Besides, it severely harms the economies of many countries. Violence and crime represent a problem that we all have to deal with. Unfortunately, while governments are trying to counteract insecurity, citizens continue to suffer from criminal violence. As an example, in Mexico, estimations based on the number of investigations initiated in 2019 indicate that the most common types of crimes include nearly 150 thousand cases of theft from 4-wheeled cars, more than 82 thousand cases of residential burglary (with and without violence), almost 83 thousand cases of robbery to passers-by on public streets, around 18 thousand cases of theft in public transportation, and over 118 thousand cases of robbery to businesses [1]. The types of crimes described above, added to the lethal records, motivate families and businesses to install security cameras in homes, parking lots, and public places, to mention some. Although crimes can occur anywhere, in this work, we focus on violent acts in public places such as streets, pavements, shopping centers, or schools.

Crime leaves psychological marks on its victims, which are unable to cope with using common psychological resources [2]. The latter causes various types of emotional sequelae that negatively affect their lives. Among the most common disorders we can mention negative feelings, anxiety, constant concern due to trauma, depression, progressive loss of personal confidence, decreased self-esteem, alterations in the heart rhythm, and insomnia [3]. Conversely, physical integrity produces other effects such as injuries, trauma, cuts, fractures, and more. Regarding

physical violence, it is possible to identify the aggressor's actions by looking at the interaction between the aggressor and the victim (their actions and movements), which are observable for just a few seconds. Observing such interactions is the basis of our proposal.

A reliable video surveillance system should be aware of what is happening at a particular moment and the actions that could result in potentially dangerous situations [4]. Covering all events in most current video surveillance camera systems requires security workers to watch what happens periodically. One of the main disadvantages of this approach is the cost of keeping people working for so long, which is likely to cause human errors because of tiredness and fatigue. Even if we can overcome these issues, each observer may rate an individual's suspicious action differently depending on their criteria and experience. These differences may cause the detection of actual suspicious actions to take longer to be labeled as such and to avoid undesirable situations [5]. Detecting violent attacks, robberies, and other criminal conducts requires a change in video surveillance systems that adapts to the current requirements [6]. For example, Martínez-Mascorro et al. [7] proposed a video surveillance system powered by AI that allows the early detection of robbery attempts through the analysis of behavior. In their work, they classify video segments as normal or suspicious, where the latter is interpreted as an indication that a robbery is about to occur. As we can observe, the constant development of artificial intelligence algorithms seems like an excellent option when detecting different behaviors in people captured by video surveillance cameras, which is the focus of this work.

Current research uses different techniques based on deep learning to tackle problems for image/video pattern recognition such as 3D convolutional neural networks [8–10], hybrid architectures using convolutional neural networks, and long-short term memory [11, 12], or including any form of deep learning for feature extraction, transformation, or classification at some point in their pipeline. However, their use may be limited because the topology of the neural networks is fixed. Therefore, the number of nodes and hidden layers must be chosen, leading to questions such as: how many hidden layers and hidden neurons per layer should the networks have? How many training pairs should the training phase use? Which neural network architecture is the best? Finding the values for the parameters is one of the most important steps because failing to do so usually causes the neural network not to be powerful enough to meet the requirements [13] or increases the error by accuracy saturation [14].

Neural network developers usually answer the above questions using their previous experience and design a fixed architecture throughout the training. On the opposite, in nature, the human brain has evolved up to 100 billion neurons and 100 trillion connections. This number surpasses, by far, the number of neurons and connections our ancestors counted with [15]. Similarly, hyperparameters in artificial neural networks should change in time by leveraging the ancestors' knowledge.

Neuroevolution is a type of artificial intelligence inspired by the evolution of biological nervous systems, used to generate artificial neural networks with a suitable topology and parameters to perform a given task. Neuroevolution has proved to be a reliable alternative in different tasks such as the classification of the electrocardiographic signal in health applications [16], the design of mobile robot controllers in robotics [17] or in video games to play at a human-like level [18]. However, detecting violent behavior in videos through neuroevolution remains an unexplored topic. Given the good results of neuroevolution in different scenarios, we consider it may also achieve competent results for generating networks that correctly classify the behavior shown in a video as violent or normal.

In this work, we are motivated by the idea that by providing tools that allow for the prompt detection of violent behavior, we can help people maintain a safe environment and respond accordingly to such stressful situations. Then, we address the problem of creating a suitable architecture for a neural network to classify behavior in videos correctly. For this purpose, we use genetic algorithms to create and evolve neural networks that, without human intervention, recognize violent-related activities in people who appear in the range of vision of a surveillance camera.

To summarize, the main contributions of this work are as follows:

(1) Advancing towards surveillance systems that rely less on human intervention since our system can adequately discriminate violent behavior from a normal one in video clips from the database used for this work.

(2) Contributing to the automation of designing neural networks since we use a genetic algorithm to generate different architectures of neural networks to detect violent-related activities in videos.

(3) Providing empirical evidence that both dense layer–based and 3D convolutional layer–based networks can sidestep the task of detecting violent-related activities in videos.

The remainder of this document is organized as follows. In Section 2, we introduce different approaches from the literature that address the detection of different behaviors, as well as related works on the evolution of neural networks. Section 3 describes the technical overview of different methods used in this work for evolving neural networks. The dataset, the procedure to extract features from videos, the fitness function, and the metrics considered in this work are detailed in Section 4. Section 5 presents the experiments and results of the proposed approach for generating shallow, deep dense-based, and 3D convolutional neural networks, respectively. Finally, Section 6 presents the conclusion and some ideas for future work.

2. Related Work

This section describes relevant works that are significantly related to the problem we address and how we solve it in this

work. We first describe the problem and emphasize how current solving approaches are limited since they depend on human intervention. Later, we describe different solving approaches described in the literature, which cover systems using spatiotemporal features, pose estimation, and deep features. We finally describe neuroevolution and NEAT as a way to familiarize ourselves with the rationale behind this work.

Intelligent video surveillance systems are based on observable behavior monitoring, categorizing it into different classes. For instance, we can mention theft, violence, and fraud detection, among others [19]. Existing intelligent video surveillance systems for behavior classification generally fall into one out of three broad groups, according to the set of features used to classify: two-dimensional, three-dimensional, and deep features. Among these categories, two-dimensional recognition systems have been explored the most [19].

Systems using two-dimensional features employ spatial information and RGB intensity to classify the cases. One example of such systems was proposed by Li et al. [20]. Their system relies on analyzing spatiotemporal video-volume space configuration to detect and localize anomalies in videos. Their method considers three main steps. The first step is the construction of an activity codebook through the extraction of low-level features referring to the global activity patterns on the video. The second step detects the anomalies at local sites using a Bag-Of-Words approach on a video cube (a spatiotemporal video window), producing composition-representation vectors. The third, and final step, builds a dictionary to detect the video activities and locates the regions where anomalies occur.

Another example using 2D spatiotemporal features involves two optical flow-based motion descriptors in Ref. [21]. The authors propose two spatiotemporal approaches to solve the detection of abnormal activities. The first one, called Silhouette and Optic Flow-based Features (SOF), uses background subtraction, then the optic flow values are generated. On the other hand, the second motion descriptor uses the Dense Trajectory Based Features (DTF), a method that calculates trajectories to create Histogram of Gradients (HOG), Histogram of Optic Flow (HOF), and Motion Boundary Histogram (MBH) descriptors, followed by a standard bag of visual words approach to create a visual vocabulary (codebook). Both SOF and HOF use a one-class Support Vector Machine (SVM) classifier to classify normal and abnormal samples. The two methods proposed endure major drawbacks despite promising results. As in most real-life scenarios, these methods can work well in a particular dataset, but their performance worsens in others. This is a problem that is present in real-life implementations. Not all the scenarios will be the same, yet the algorithm must generalize to be distributed in different environments and work acceptably.

Methods employing two-dimensional features typically use speed, direction, trajectory, and optical flow to understand the behavior of the entire scene captured by a camera. However, these characteristics apply to the people in the scenes and the elements that move in the videos. Thus, objects falling, rolling, or other elements that work by moving on their own can cause noise in the evaluation of scenes. Moreover, changes in lighting, reflection, or background clutter can significantly affect the method's performance. One way to better understand the movements relies on skeleton-based features. These approaches tend to have better results because they are less limited than previous ones. It is better to use the shape of the poses as well as their deformations over time to understand the silhouette of a person similar to real life, rather than just the representation of the two-dimensional segmentation approach that would consider a binary image silhouette [22].

Therefore, the pose estimation approach is gaining strength among researchers despite the need for better hardware. For example, Markovitz et al. [23] interpret people's poses using graphs to mitigate the viewpoint and lighting problems of the scenes. Weighted adjacency matrices generate pose graphs where each node represents a key point, a body joint, and each edge represents some relationship between two nodes. Some implementations work with skeletons at the cost of using additional devices. For example, Chaaraoui et al. [24] use Microsoft's Kinect since this type of RGB-D device has become cheap over the years and offers remarkable results in retrieving information from human bodies. They propose a novel skeletal-based spatiotemporal feature called the Join Motion History feature (JHM), representing the 3D location of skeletal joints and motion's age. As a classifier, they use the Bag-Of-Key-Poses method described by Chaaraoui et al. [25].

To conclude, deep learning models produce classes based on deep features. Such deep learning models include convolutional neural networks (CNNs) and recurrent ones, which tend to have more embracing use in addressing problems of this kind. One example of such methodologies is described by Vignesh et al. [26], where a CNN is used together with Long short-term memory neural networks (LSTM) to classify between normal and violent behavior frames. The convolutional network learns the spatial features of the images, and the LSTM neural network learns the long-term dependencies between frames. The main disadvantage of this work is the extraction of spatial and temporal features as different models carry it out. Li et al. [8] use a 3DCNN model without the addition of handcrafted features or RNN architectures to avoid the need for two different models. Handcrafted features can be eliminated because methods focused on deep learning can yield robust results and great accuracy.

The works described above are built on hand-designed architectures. In most cases, the hyperparameters are defined and set based on the expertise of their designers. Various works based on genetic algorithms have dealt with the constructions of neural networks by minimizing human intervention. One example is described by Xie and Yuille [27] as a genetic CNN. They propose a direct encoding represented as a fixed-length binary string that, in turn, is split into stages trying to simulate the parts into which state-of-the-art models are partitioned. The most critical limitation of genetic CNN concerns the genome representation since a fixed-length encoding affects networks that need to

be deeper in order to converge to global (minima/maxima) values of the solution space. Besides this, the kernel size is also fixed within each stage. This limitation can help prevent networks from growing out of control and restrain the search space's size.

Large-Scale Evolution of Image Classifiers [28] overtakes the disadvantages of the previously detailed approach with fewer restrictions in depth, arbitrary connections, and numerical parameters by relying on individual encoding as graphs called DNA. The authors predefined a set of operations, similar to human designers' actions when constructing a network, which can be carried out on the mutation step as Remove-Convolution, Alter-Stride, Filter-Size Alter-Number-of-Channels, Add-Skip, and more. Authors of Large-Scale Evolution of Image Classifiers emphasize the massively parallel computational power required by their work. Nevertheless, some works seek to optimize resources. For example, Efficient Multi-Objective Neural Architecture Search via Lamarckian Evolution [29] proposes a multi-objective evolutionary algorithm for architecture search. This multi-objective function allows an approximation to the Pareto front (a front of solutions dominating all other solutions), optimizing several objectives such as accuracy, size of the network, and number parameters. To avoid large resource consumption, the authors choose individuals that most fit the Pareto front using the set of cheap objectives (easily calculable objectives such as the number of parameters) and evaluating this subset on their expensive objectives, in addition to employing a Lamarckian inheritance where learned parameters are passed to network's offsprings.

Initial approaches for neuroevolution are based on direct coding. The genome treats each node and connection as an individual element, making it more difficult and memory-consuming to store all of them in the population during the evaluation of generations. Another disadvantage of this idea is that it cannot reuse elements and must be coded and evolved independently. The Hybercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) [30] uses Indirect Encoding employing a variant of an artificial neural network called a connective Compositional Pattern-Producing Network (CPPN) [31], which represents sophisticated repeating patterns in the Cartesian space. These CPPNs produce spatial patterns based on functions such as Sigmoid, Gaussian, Sine, Cosine, Tanh, Relu, and others to exhibit properties of all these activations and create a symmetrical output, a repeating pattern, a repeating pattern with variation, and more. These patterns can be produced thanks to spatial interpretations of patterns generated within a hypercube (hence the name) as connectivity patterns in a lower-dimensional space. The CPPN does not work as a neural network for inferring based on the input data. Instead, it generates the weight of the connection between two neural network nodes being searched. It receives the coordinates of the Cartesian space between two nodes located in an n -dimensional space called the substrate.

However, HyperNEAT also presents some disadvantages. The most obvious one concerns the positions the hidden layer nodes should have. This is a decision that must be made by the developer as CPPN is not able to determine them, although there is an extension to HyperNEAT called ES-HyperNEAT (evolvable-substrate HyperNEAT) [32] which looks for areas with much variance in the pattern produced by CPPN, and it remains costly in execution time.

In this work, we propose using NeuroEvolution of Augmenting Topologies (NEAT) and its extension, CoEvolution Deep NEAT (CoDeepNEAT), to solve the task of detecting people comparing two different types of networks: shallow and deep ones based on dense and 3D convolutional layers. These networks were selected as they do not carry over other approaches' problems and solve previously untackled problems (like the co-evolution of different niches).

3. A Brief Review on NEAT

The literature describes several works on the creation of neural networks through an evolutionary process [33, 34]. Such works can be divided into two large groups based on the encoding they use [35]. The methods that rely on direct encoding use the genome to establish the network topology within the phenotype. Usually, the representation relies on binary encoding to represent the neural network connections. However, this method is limited to the size of the matrix used to store the bits [34]. The second group is formed by the methods that use indirect encoding. These methods rely on rules to precisely indicate how the phenotype—that is, the neural network—will be constructed based on such rules. The most significant problem with this type of encoding is that it does not directly map the information from the genotype into the phenotype (the neural network), which may move the search away from desirable solutions. In this work, we evaluate two different types of neural networks, Shallow networks generated by NeuroEvolution of Augmented Topologies (NEAT) and Deep networks evolved by CoEvolution Deep NEAT (CoDeepNEAT). The overall technical explanations of these techniques are given as follows.

3.1. Neuroevolution of Augmented Topologies (NEAT). Genetic algorithms have applications in many fields, including Neural Networks. NeuroEvolution of Augmented Topologies (NEAT) [36] has been used for reinforcement learning problems and has even proven to perform better than other reinforcement methods like Adaptive Heuristic Critic for solving the single-pole balancing problem [37].

Using a direct encoding for evolving neural networks is difficult because of the different components involved in an artificial neural topology, such as the number of neurons and connections. Instead, NEAT encodes each genotype with two lists, one for storing all the nodes (called *node genes*), which includes inputs, hidden and output nodes in the neural network, and a second one (*connection genes*) that represents all the connections between every pair of nodes as seen in Figure 1. Each of the nodes in the node genes has two attributes, one node number and the other for the type of node it represents (which can take a value from *Sensor* (Input),

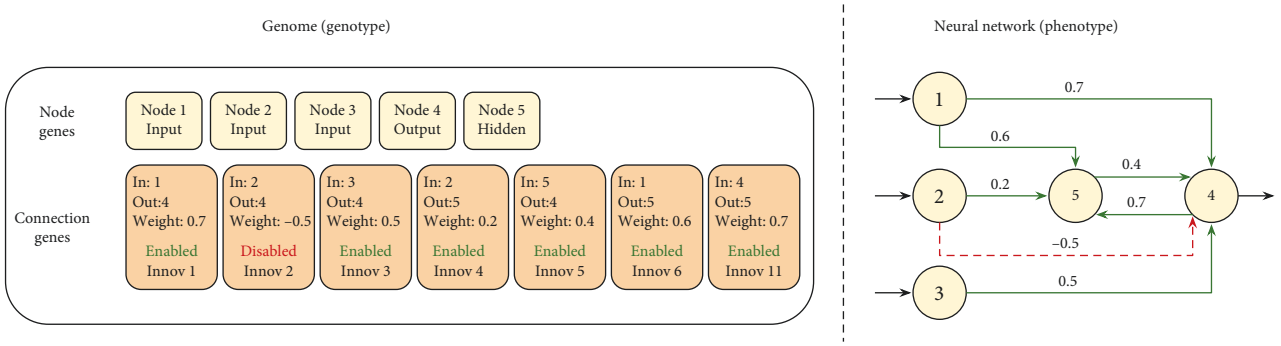


Figure 1: Individual representation NEAT: genotype and phenotype.

Output, and *Hidden*). On the other hand, the elements in the list of connection genes have more than two attributes specifying the in-node, out-node, the weight of the connection, a Boolean pointing if it is enabled or not, and an innovation number allowing finding its corresponding genes. A visual representation of these genotypes and phenotypes is shown in Figure 1.

The innovation number of each element in the list of connection genes allows NEAT to use information in the historical origin to perform crossover. When two networks are selected for crossover, they are lined up following the gene's historical markings producing *matching genes*, and *disjoint genes* referencing nodes and connections being removed or kept in a new neural network. Moreover, mutation and crossover could lead new individuals' fitness to drop and lead them to a lower probability of survival even when these mutations can significantly help future generations. This is the reason why NEAT uses *speciation* acting like a niche, where individuals are only allowed to compete with others in the same niche.

3.2. Coevolution Deep NEAT (CoDeepNEAT). Just like HyperNEAT, CoDeepNEAT is also another extension of NEAT [38, 39]. NEAT is first extended to deep neural networks by calling it "DeepNEAT" using the same principles: individuals of minimum complexity are represented as graphs, nodes and edges are added or removed using mutation, and historical markings are used during crossover to align individuals and combining them among the most similar ones, the population is divided into species based on a similarity measure and each species grows proportionally to fitness with evolution occurring separately in each species. The main difference to NEAT is that it does not evolve complete networks; each individual is represented as a graph, and each node represents one or more layers of the deep neural network. In addition to these layers, each node also contains a table with hyperparameters, known as "node hyperparameters," containing real or binary values mutated through evolution via a random bit-flipping (using a uniform Gaussian distribution) depending on the type of value. Those also help identify the type of network (convolutional, fully connected, recurrent) and properties such as the number of neurons, kernel size, and activation function. The edges no longer indicate weights but with which other layers they are connected to.

The graph also has a table called "global hyperparameters" that, as the name suggests, describes the hyperparameters that apply to the entire network, such as training algorithm, learning rate, and data preprocessing steps. DeepNEAT tends to produce complex and unprincipled structures, and CoDeepNEAT solves this by evolving two different populations of blueprints and modules separately.

Blueprints refer to graphs representing individuals where each node points to one subpopulation or species of modules. Each module is a graph representing a small DNN. When the evaluation of a generation starts, the blueprints replace the node with a module to the species it points to, as illustrated in Figure 2, and the hyperparameters in the "node hyperparameters" table are applied. Once the network is assembled (when all the nodes were replaced by their modules), the "global hyperparameters" table is also applied, and the evaluation starts. The overall fitness of the network is attributed back to blueprints and modules as average fitness. This evaluation reduces noise and allows blueprints or modules to be preserved for future generations even when they performed poorly in an assembled network. Once CoDeepNEAT ends, the best network is trained until it converges and is evaluated using a different test set.

4. Dataset, Features, and Metrics

Features used to feed the networks are an essential part that should be carefully chosen as they help the models to converge faster with the best inference results. The dataset and how the features were obtained from the videos are described in this section. Moreover, the fitness function (how well an individual performs during the training) and the evaluation metrics (how well each model performs on unseen data) are also described in their corresponding subsection.

For a reference, Figure 3 provides a general view of the pipeline proposed for evolving shallow and deep neural networks followed in this work.

4.1. Kranok-NV Dataset. This work uses the Kranok-NV dataset to evolve and train neural networks. The Kranok-NV dataset [40] consists of 2,026 videos, divided into 597 violent behavior videos and 1,429 normal behavior ones with a

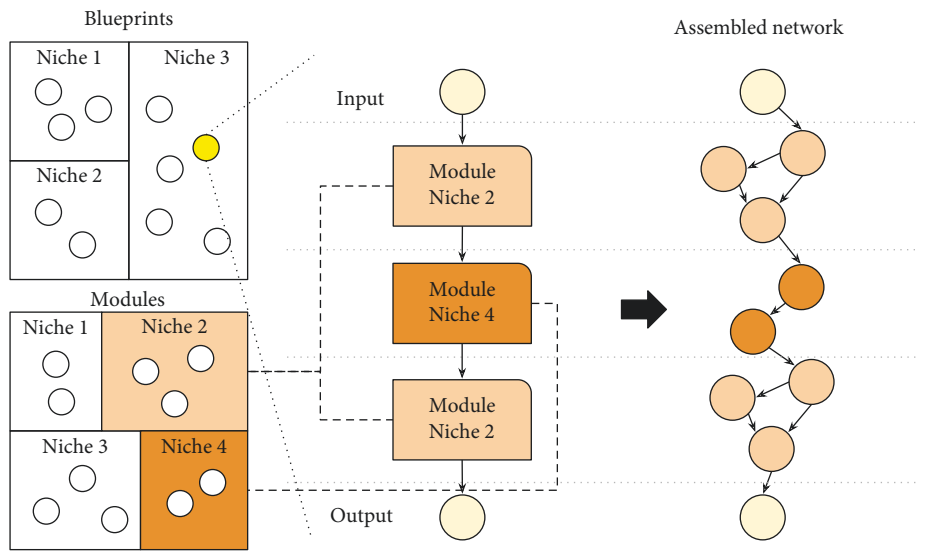


Figure 2: Coevolution of blueprints and modules.

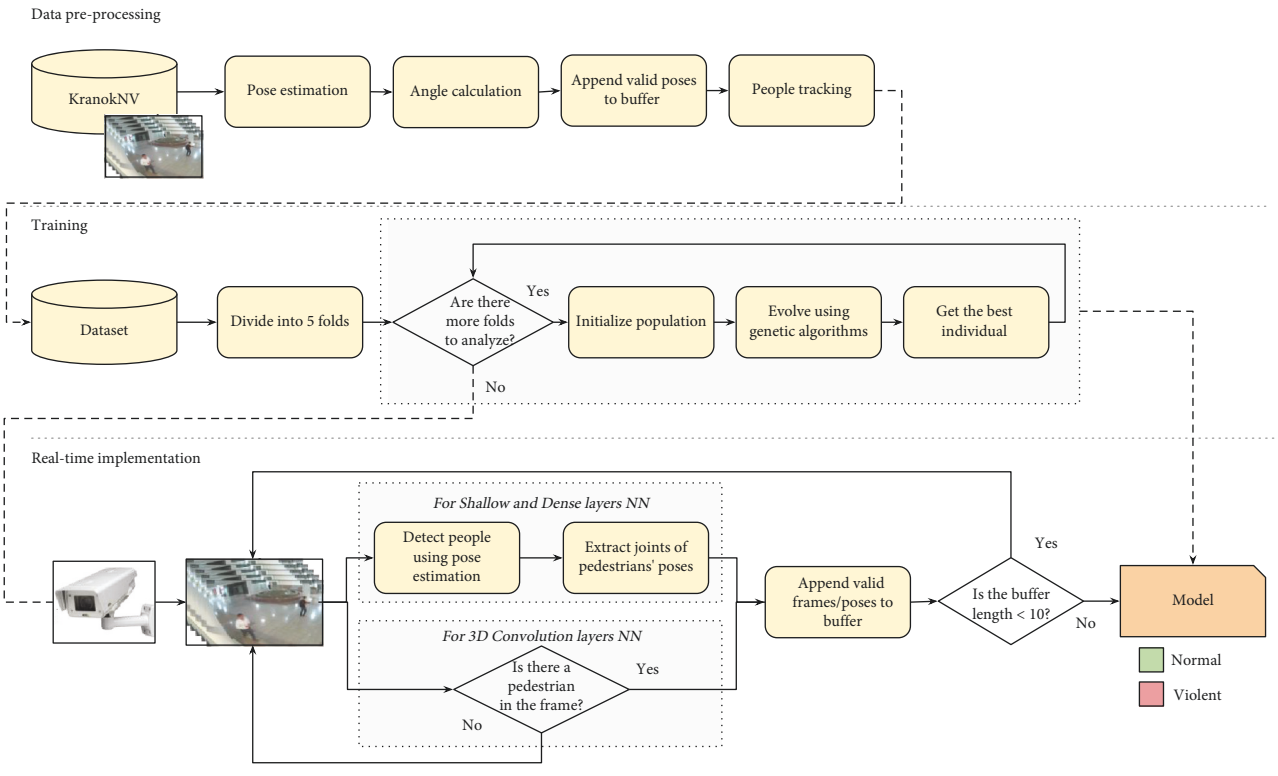


Figure 3: Pipeline for evolving shallow and deep neural networks.

resolution of 840 pixels width and 472 pixels height, each having a different duration. This dataset was built for explicit use for violence detection in classification tasks. Violent behavior videos were made up from the YouTube website of people training in different practices such as boxing and kickboxing in joint or individual activities. In contrast, normal behavior videos consist of people going from one place to another walking down a corridor from three different cameras/angles in a closed-circuit television camera system. There is a noticeable class imbalance in the number of videos in the dataset. However, violent behavior videos have more frames as a whole than normal behavior videos because of their length. The class with fewer instances (normal behavior) was upsampled using two different techniques: zoom and mirror. The camera's vision field gets smaller by using zoom but makes objects (including people) bigger. Mirror comprises a horizontal flip in the image leaving, for instance, people that went from left to right now

going from right to left. In addition, 5-fold Cross-Validation was used in the training phase partitioning the dataset into five equal-sized subsamples trying to keep as close as possible to 50/50% class balance in the number of frames. Kranok-NV is freely available at https://www.kaggle.com/kevinbkwanloo/kranoknv.

4.2. Feature Extraction. Previous works not based on neural networks have based their solutions in features like binary image silhouette [22] or in histograms of gradients [41], among others. This has been done regardless of whether such features represent people and may represent other noisy elements. Furthermore, those features do not directly represent attributes for behavior classification. Instead, they tend to represent the entire scene. Thanks to machine learning advances, human pose recognition has also had advances in different areas, even going from 2D output of human pose estimation in images to 3D shape–based on RGB-D devices and estimation from video. In contrast to the former described features, poses map actions of people that humans can easily interpret, so a network is expected to do the same.

There are many types of research related to pose estimation in which the detection and estimation of people's poses are made in a short time with high accuracy. In this project, such a task relies on deep learning models. Deep learning models for human keypoint detection are classified into two groups, depending on the order of the operations of the approach; top-down and bottom-up, bottom-up models are more robust to occlusion and complex poses. For instance, OpenPose [42–45] is defined as a *multi-person real-time keypoint detection* and operates as a bottom-up approach and was (and continues to be) one of the best alternatives for pose estimation. Nevertheless, it requires high-end GPU hardware to run correctly at competent Frames Per Second in live videos, like those in closed-circuit television environments. Fortunately, there are other options, such as Lightweight OpenPose [46], which is the one used in this work. The authors designed this option intending to maintain the excellent inference results but make it more suitable for real-time performance on edge devices, not only to be run in a GPU but also usual CPUs.

Two out of three neural networks evolved in this work share the same features. Shallow and deep neural networks based on dense layers expect numerical values as input in their input layers, leading to the same data for training and testing. In contrast, 3D convolutional neural networks expect images represented as numerical values in a 3D vector (height, width, and channels), requiring a particular feature preprocessing step. Nevertheless, a single pose cannot provide enough evidence about people's behavior considering a person could have a pose similar to another different action and be classified as such. We can use the poses' keypoints as spatial features plus their evolution throughout frames to extract the context as in Figure 4. This is achieved by stacking the poses from previous frames and iteratively keeping track of them. The angle between keypoints determines the spatial features. This is utilized to avoid using the raw Lightweight OpenPose's keypoints output and help offer a better generalization in the features. Such angles are calculated by using the arctan formula.

On the other side, temporal features are added by taking the person's pose in the video and seeing how the pose evolves in time. The longer the period, the more complex the neural network should be. Nevertheless, shortening the length of the period to reduce the network complexity may seriously decrease its classification power. The process for tracking people is described in more detail in Section C. Thus, measuring the relations in the people's poses and tracking their change over a short period (ten frames) can perfectly describe their behavior and be mapped to a classification using a neural network.

All the preceding describes the features used to input shallow and dense neural networks. Nonetheless, the feature extraction for 3D Convolutional Neural Networks (3DCNNs) is simpler. This is mainly attributable to 3DCNNs having the skill to model complex, both spatial and temporal, characteristics using cube-shaped kernels [9]. Experimentally, 3DCNN should outperform models with handcrafted features as long as the feed sequences are of correct data related to the task to be solved. Therefore, only video sequences with people in the field of view must be used in training and real-time implementations. Consequently, noise in the training data refers to every sequence not containing people, primarily if that sequence comprises objects in movement, although frames with no people can be labeled as nonaction.

3DCNN receives a 10-frame sequence as a context span to maintain a similarity in time concerning the input of shallow and deep dense networks. As previously stated, only if there is at least one person in the frame, then that frame can be used as part of the sequence. However, if there are no people in the frame, it cannot be used as part of the sequence nor past frames where people did appear. Thus, the entire sequence is discarded. Both features, pose angles based on pose estimation and video sequences, use ten frames as the pose's context, and networks do not receive noisy frames without people. In conclusion, the algorithm somehow trains all the networks with the same features.

Overall, a brief description of the previously detailed feature extraction step is detailed in the data preprocessing stage depicted in Figure 3.

4.3. Tracking. Tracking employs the keypoints of the people's poses to simulate the skeleton as independent variables to feed the networks. As mentioned before, in this work, we use Lightweight OpenPose [46] as a pose estimator. The neural networks should classify human actions based on their pose with such a skeleton. Conversely, a single pose as the context may attribute multiple false-positive classifications given that it could match with another pose of a different activity as someone moves. Thus, additional context is essential and should also be fed into the network and the pose. The temporal features are extracted by concatenating a sequence of poses as long as the person appears in the video sequence (tracking).

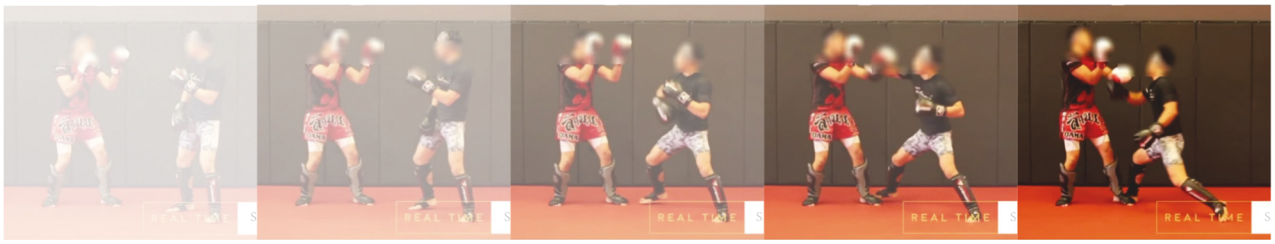


Figure 4: Poses' context from time.

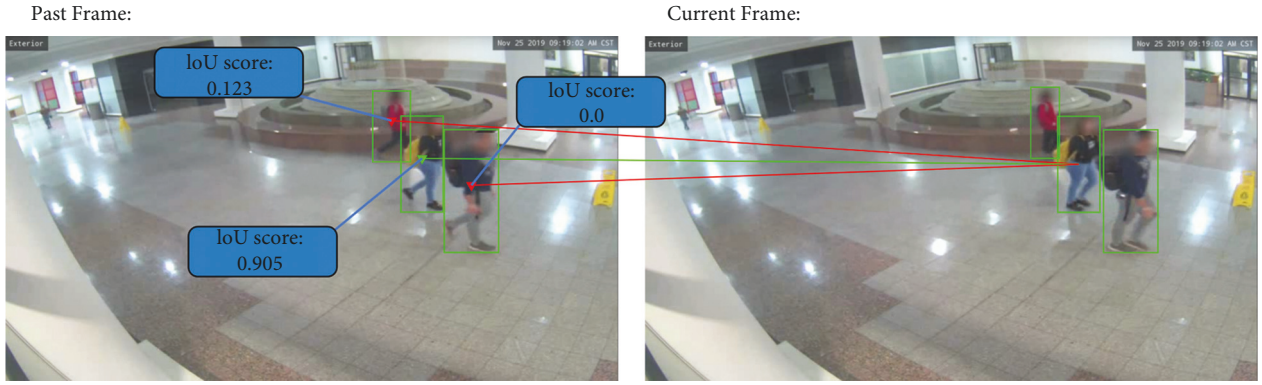


Figure 5: Example of IoU applied in videos.

The approaches for detecting and tracking people are usually separated into two categories: Model-Free-Tracking (MFT) and Tracking-By-Detection (TBD). Tracking-By-Detection trackers are more widely used nowadays. Such models have several advantages, such as not requiring people in the first video frame, dealing with many people in the image, high detection accuracy, and not accumulating minor errors resulting in significant performance degradation in long videos. The solution comes by localizing the target in each frame without reliance on previous inferences. Instead of adding a new model on top of our solution approach, we decided to perform the tracking task using only the key-points extracted by Lightweight OpenPose. The operation works by associating each person based on their whole body comprising joint points with a unique identifier on the first frame the person appears in the image. On posterior frames, each individual is compared with every person in the previous frame using their area of interest/bounding box by an operation called Intersection-over-Union (IoU), chooses the one with the highest rate, obtains its ID value, and assigns it back as its current ID will allow us to track their poses over time. Figure 5 shows a visual example of applying IoU.

4.4. Fitness and Evaluation Metrics. Fitness must be assigned to each individual in the population over generations. Although we could have chosen the model's accuracy as a fitness value, it does not consider the class imbalance, and then, it does not penalize large errors in the classification. Fortunately, Shallow, Deep Dense, and 3DC neural networks use the same loss function, which does take large errors into account. Categorical Cross-Entropy was chosen as such a loss function given this task is a classification task using a one-hot encoded output.

We have opted for five popular metrics used in many classification applications. These metrics are defined below.

$$\begin{aligned}
\text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}},\\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}},\\
\text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\
\text{F1} &= 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}},\\
\text{Specificity} &= \frac{\text{TN}}{\text{FP} + \text{TN}},\\
\text{Balanced accuracy} &= \frac{\text{Recall} + \text{Specificity}}{2}.
\end{aligned} \tag{1}$$

In all cases, TP, FP, TN, and FP stand for "true positives," "false positives," "true negatives," and "false positives," respectively. All these values can be extracted from the corresponding confusion matrices. It is relevant to recall that the values for Precision, Recall, and F1-score obtained from a confusion matrix when violent behavior is the class of interest may change when the class changes to normal behavior. Conversely, the accuracy and balanced accuracy do not change when the class of interest changes.

5. Experiments

In this section, we describe the experiments conducted and the results obtained.

5.1. Evolution of Shallow Neural Networks through NEAT. This section describes the NEAT implementation and how we incorporated it into the training of the shallow neural networks. We have selected a set of hyperparameters that produce an acceptable performance of the genetic algorithm on each fold. These hyperparameters were obtained by preliminary experimentation, running the algorithm several times until finding the ones that perform satisfactorily on different scenarios.

It is well known that, for genetic algorithms to work correctly, they require to tune many hyperparameters, such as initial population size, initialization of individuals, probabilities for mutation, crossover, and more. A slight modification of these may lead to drastic changes, for better or for worse. Several runs of the NEAT-based part of this work were conducted to find the best configuration of hyperparameters for producing the neural networks. The selected configuration considers running for 100 generations, where the probability of adding or deleting a connection between nodes is 50%, respectively. Besides, the probability of adding and deleting a node was 50% for each case. Lastly, only the two best individuals from each species remain untouched and are moved to the next generation, while 20% of each species is allowed to reproduce.

When an individual is initialized, a connection between input and output nodes is established, with a 50% probability. This means that, on average, each individual starts with half of its input nodes connected to half of its output ones. The former was done to avoid starting with completely disconnected neural networks (individuals), which may increase the algorithm's convergence time. Finally, the neural network has two output nodes representing violent/normal behavior, with a softmax postprocessing. The results presented in this section correspond to one network out of the 100 produced (the one with the highest fitness) using ten frames as input.

Figure 6(a) presents the confusion matrix of the Shallow neural network when violent behavior is the class of interest. 245,874 (48.95%) samples were correctly classified as violent behavior, while 221,750 (44.15%) samples were correctly classified as normal behavior. The larger the TP and TN in a confusion matrix, the better the network's performance (they represent the successes). Thus, the Shallow neural network performed notably, obtaining an overall accuracy of 93.6741%.

As we mentioned before, the values for Precision, Recall, and F1-score depend on the class of interest. Although both types of behavior are relevant, it is critical to identify violent behavior over the normal one in this work. Table 1 shows the results of Precision, Recall, and F1-score for both violent and normal behavior. Although the performance is similar in all the metrics, for Precision and F1-score, the Shallow neural network was slightly better at identifying violent behavior than the normal one.

As observed from Table 1, the Shallow neural network seems to exhibit a well-balanced behavior. A balanced accuracy of 93.7111% confirms this. Overall, the Shallow neural network performed well for classifying the video samples as violent or normal.

5.2. Evolution of Deep Neural Networks through CoDeepNEAT: Dense Layers. The following model is an extension to evolve deep neural networks to detect violence in videos. There is a similarity between the shallow neural network evolved using NEAT, and the dense, deep neural networks evolved using CoDeepNEAT: how input values are fed into the networks. Both models expect feature vectors representing ten frames of angles between joints from the body of people in videos; therefore, both models use the same dataset for training and testing along with 5-fold cross-validation.

The hyperparameters for the macro and micro-architecture, i.e., the Blueprints and Modules, respectively, were selected after several runs and are described as follows. The blueprints' hyperparameters are set and not modified throughout the evolutionary process. Such hyperparameters include the loss function using *categorical cross-entropy*, Adam optimizer, along with its learning rate with a fixed value of 0.005. The output layer activation function is softmax right as in the shallow model approach described in the previous experiment. The only parameter we can vary is the number of nodes in the blueprint; it was set to three as a maximum and one as a minimum.

The modules' parameters (dense layers that will replace the blueprint nodes) comprise some nodes in a layer, in a range between 35 and 80, and activation and function, either ReLU or Sigmoid. In addition, each module can have a maximum of three dense layers and a minimum of one. In this manner, considering that the blueprint could have three modules as maximum and each module three layers, the largest neural networks produced by the algorithm would contain nine hidden layers. Nonetheless, the depth is extended by placing fixed layers at the beginning (after the input layer) and the end (before the output layer) of each blueprint. These fixed layers are used by reasoning that layers should have fewer nodes the closer to the output layer. Such fixed layers do not belong to any niche module. Instead, they are placed in their respective places at the beginning of the evolution process when the blueprint population is created and not removed or modified. One of these two "fixed" layers is contained after the input layer and can have node units ranging from 150 to 200 using a Sigmoid or ReLU activation function. The second "fixed" layer follows the previous one and can have node units from 80 to 120. All the modules (evolved in niches through the generations of the genetic algorithm) come after this second fixed layer, allowing the network to grow initially and eventually become more condensed towards the output. Furthermore, each dense layer in the modules is followed by a dropout layer using a rate from 0 to 0.07 units to drop. This helps the

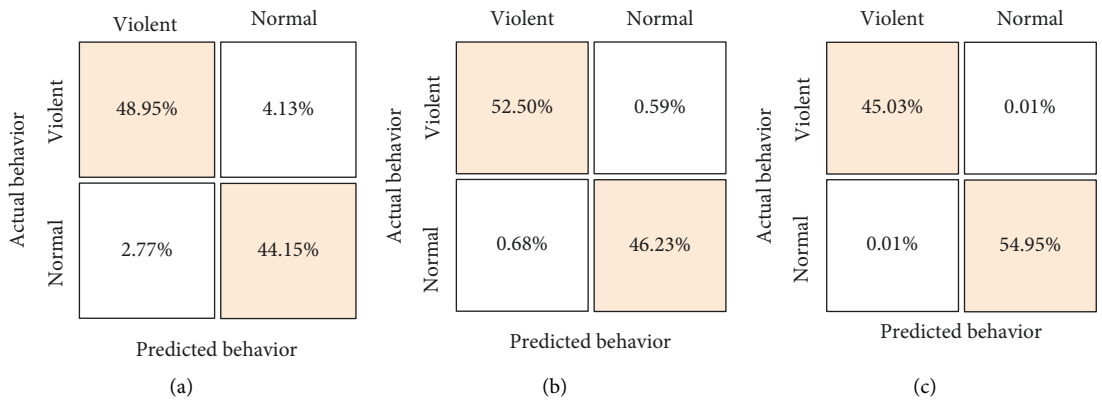


Figure 6: Confusion matrices for the three networks evaluated. (a) Shallow neural network. (b) Deep dense neural network. (c) 3D Convolutional neural network.

Table 1: Precision, recall, and F1-score obtained by the Shallow neural network for both classes of interest (violent and normal behavior).

	Precision	Recall	F1-score
Violent behavior	94.8773	93.1110	93.9859
Normal behavior	92.3655	94.3113	93.3283

Table 2: Precision, recall, and F1-score obtained by the dense neural network for both classes of interest (violent and normal behavior).

	Precision	Recall	F1-score
Violent behavior	98.7226	98.8937	98.8081
Normal behavior	98.7456	98.5520	98.6487

network to prevent from growing so much and producing overfitting.

The genetic algorithm runs for 20 generations using a population size of 15 blueprints and 30 modules, both evolving within their related niches. When possible, niches group similar blueprints and modules in three different species using the k-means clustering algorithm. Evolutionary parameters cover mutation set to 50% probability, crossover rate of 20%, elite modules and blueprints of 20%, the latter referring to keeping a portion of chromosomes intact and passing the individuals to the next generation without modifications.

Fitness is assigned when the blueprint's nodes are replaced with the respective modules niches they point to. One thing to look out for is that training each network using the entire dataset in every generation is computationally expensive and time-consuming. Therefore, only a portion of the dataset's training and testing data are used. For the fitness assignation, the networks are first trained using five training epochs with randomly picked samples equivalent to only 40% of the training data, and the fitness is yielded by the loss using 30% of randomly picked samples from the test set. Finally, the algorithm outputs the networks with the highest fitness and is converged using the entire training set during 15 epochs; its parameters are saved, and the classification results are presented in the section below.

As can be seen in Table 2, the metrics of the evolved networks using 5-fold cross-validation for ten frames as inputs outperform those from the Shallow neural network developed in the previous experiment. The deep dense neural network performs slightly above 98% in the same metrics used to evaluate the shallow network. From the confusion matrix depicted in Figure 6(b), we can observe that the overall accuracy has increased to 98.7333%. Moreover, the value of 98.7228% in the balanced accuracy confirms that this neural network maintains its quality across the two classes of interest.

5.3. Evolution of 3D-Convolutional Neural Networks through 3D-CoDeepNEAT: 3DCNN. So far, two types of neural networks have evolved: shallow and deep based on dense layers (also known as fully connected layers) using NEAT and CoDeepNEAT. The input values of these two types of networks are temporal features that indicate how the angles between joints change in a span of ten frames. However, the spatial features must also be considered to classify the activities. Convolutional neural networks (CNN), also known as Shift Invariant or Space Invariant Neural Networks, are highly used for this task, considering they are skilled at recognizing patterns in the input images by producing abstractions that are visible in activation maps. In short, the input images are feed representing low-level features, and the CNN is responsible for obtaining high-level features as data travel across the network.

CoDeepNEAT was developed to evolve convolutional neural networks. In this work, we have extended its functionality. We have seen that CNNs are better than fully connected and shallow neural networks for extracting the features from an image, although an iteration of CNNs adds one more dimension called 3D convolutional neural networks. As the name suggests, 3DCNN convolve cube-shaped kernels in a set of given images transforming their input into four dimensions; height, weight, number of channels, and sequence length. The sequence is conformed of n images stacked one after the other. The addition of images in the

input eliminates the limitations of previous 2D CNN that convolve two-dimensional kernels into single images.

For the previous reasons, CoDeepNEAT was expanded to evolve 3DCNN and see if it can generate suitable architectures for these more complex networks. With that, the model addresses the same questions that arose with the previous models: how do we know that we have selected the architecture that gets the most out of our training data while at the same time generalizing to perform on previously unseen data?

Again, by using 5-fold cross-validation, the genetic algorithms will produce different models trained in different datasets, which will have unbiased error estimates on the test data. Many tests were conducted using a single fold from the five generated to select the most suitable evolutionary algorithm ratios such as crossover, mutation, number of elite individuals, generations, and the values of the hyperparameter tables of the nodes.

The definition of the final set of hyperparameters starts with the hyperparameters table, i.e., the blueprints. The table got the best results when the number of nodes in each blueprint was a minimum of 1 and a maximum of 2; this is the only value in such a table that can vary during the algorithm's run. Moreover, categorical cross-entropy was used as a fixed loss function, given that we are still working with the same multiclass classification problem. Adam optimizer is another fixed parameter with its learning rate established at 0.005. The softmax activation functions in the last layer of the neural network converting the values to a probability distribution. Lastly, the size of the input consists of 10 frames.

On the side of the modules, several values change in a uniformly random fashion. For example, the first *fixed* convolution layer (not included in modules) has as variable hyperparameters the number of filters/kernels that can range from 24 to 32 and the activation function deciding between either ReLU or Sigmoid. It is worth pointing out that it is the only layer that can have a max-pooling 3D compliment because arbitrary connections between layers are allowed, which means that some can be connected to more than one hidden layer. The problem emerges when some hidden layers get input values from more than one layer, which would cause incompatibility in the size of the inputs if the data come from a layer performing max-pooling while others do not. The output is concatenated to be compatible when two hidden layers point to a single layer.

Moving onto the hidden layers, they consist of numerous filters that range from 32 to 64. These filters use kernels of sizes 1, 3, or 5, with activation functions ReLU or Sigmoid. The layers' complement is dropout with a ratio from 0 to 0.5, avoiding overfitting, similar to the dense layer model evolved before.

The blueprint also contains two dense layers at the end of the architecture to help in the classification. A final dense layer is added after the nodes of the blueprint, containing between 490 and 600 nodes, with ReLU being the only activation function it can select from. The second dense layer is the last and is in charge of classifying if the behavior is normal or violent. The maximum number of layers for the modules is 2, while the minimum is 1. Thus, the maximum number of nodes in the blueprint plus the maximum number of layers in the modules entail networks with a max depth of four layers, without counting the fixed layers at the end and add-ons of each layer.

The evolution parameters are ten generations and a population of seven individuals. The number of blueprints and modules is seven too. Blueprints and modules are separated into three different species, each using k-means unsupervised classification algorithm. The crossover and mutation rates were set to 20% and 50%, respectively. Also, 20% of the population was allocated as the elite to keep the blueprints and modules without changes from one generation to another. The networks are trained using only 40% of the training data for four epochs, while 30% of the test data assign the fitness. The testing and results section presents the findings on the evolutions using 5-fold cross-validation with 10-frame inputs.

It is essential to highlight that the 3DCNN is a classifier of the whole video image which means it does not designate a probability rate to each person in the image. The entire frame (taking as context the previous nine) is classified as violent when the disturbance starts, opposite to the shallow and dense layer-based network models where each person was classified based on their own behavior and pointed using bounding boxes. The results in Table 3 confirm the superiority of the 3DCNN, supported by an overall performance of 99% on all the metrics.

Finally, we can conclude that the 3DCNN produced obtained almost perfect balanced accuracy (99.9850%), which confirms that it is a suitable model for classifying violent and normal behavior in the videos considered for this work.

5.4. Discussion. The work presented in this document shows how the coevolution of blueprints and modules also works for evolving neural networks it was not intended for (convolutional neural networks and long-short term memory networks). This idea leads to a belief that coevolution may also produce hybrid architectures. In addition, allowing arbitrary connections between layers shows that features extracted in previous layers can also help in attribute transformation on further layers. The previous practice is probably the most challenging part if a developer wants to build an architecture with skip connections at hand because one is unsure which intermediate layer would connect to the other.

However, as seen in the accuracy of the best individuals and population in CoDeepNEAT, the best individuals start with high accuracy since the beginning of the generations, which raises the question: is the genetic algorithm helping to find the best architecture? Cannot we just choose the best individual from the first generation avoiding the time of evolving the population? The short answer, according to our experiments, is no. The evaluations are performed using a different small portion of the dataset of each generation, so the initial architectures may fit that data well but may not do so well on the entire test set. The minor improvements of the

Table 3: Precision, recall, and F1-score obtained by the 3D convolutional neural network for both classes of interest (violent and normal behavior).

	Precision	Recall	F1-score
Violent behavior	99.9773	99.9886	99.9830
Normal behavior	99.9907	99.9814	99.9860

best network (and the population in general) through the generations help generalize appropriately on the test set. Moreover, at the end of each run, almost all the individuals in the population reach the global optima, which confirm that the genetic algorithm is working properly. This leaves us with the possibility of choosing the final network according to different criteria; the network with fewer parameters, the deepest network, or simply the one with the best accuracy.

The results also show that even a shallow network performs well in this task. These three models leave us with various options to choose from, according to different needs. When the computational resources are scarce, we can sacrifice some accuracy and go for a shallow neural network. As the resources become available, we can move to dense layer neural networks or 3D convolution-based neural ones.

6. Conclusion and Future Work

This work proposes using neuroevolution as an alternative to generate topologies and choose the best set of hyperparameters for a neural network to classify behavior as either normal or violent in a surveillance system. The comparison of results leverages the model's ability to evolve networks based on their depth and the type of input they receive.

Genetic algorithms can select the correct architecture of a neural network to detect violent behavior. These algorithms can produce shallow and deep neural networks based on dense and 3D convolution layers. All the models analyzed in this work can perform this task with high accuracy. However, a deeper network can extract more information from the input data, leading them to produce the best results. Although the genetic part of the process finds the best architecture, one limitation is that we still need to deal with the parameters of the genetic algorithm itself, with some of them being difficult to choose. The list of hyperparameters in neuroevolution includes the crossover, mutation, elite individuals rate, desired number of niches, number of nodes per layer, number of kernels per layer, activation functions to choose from, and more. Of course, the length of the hyperparameter catalog increases for deep learning neural networks compared to genetic algorithms. Still, it is ideal to continue looking for the best set because GAs search several parts of the solution space simultaneously, so a bit of change in the probabilities of any other hyperparameter may affect the entire population, implying stagnation in local optima.

Regardless of our efforts, there are some areas related to the neuroevolution of neural networks for behavior classification in videos that we have not explored and represent some paths for future work. For example, we can mention handling different objective classes. Instead of just classifying two types of behavior (violent and normal), future developments could focus on suspicion, protest, fainting, or harassment, to mention some. In addition, more work is needed regarding the dataset. Nowadays, the Kranok-NV dataset is limited to violent and normal behavior only. In order to explore the classification of other behaviors, we require to extend the dataset to incorporate the corresponding samples. Besides, future work may involve running the same experiments using a different, larger video dataset. In these future experiments, some hyperparameters may change on both the genetic algorithm and the neural networks.

Future work can also integrate a multi-objective search if accuracy is not the only target function. The multi-objective search would allow an approximation to the Pareto front (a front of solutions dominating all other solutions), which can optimize several objectives jointly. As a result, the list of objectives can be quite diverse, ranging from searching for the network with the lowest loss, the highest accuracy, the network with the lowest number of parameters, and estimators of the network's depth.

Finally, we are aware that we need to compare our approach against others from the literature. In this work, we compared three neural network models, all generated through their particular implementation of a neuroevolutionary approach. Our models were generated on Kranok-NV, which undoubtedly influences the outstanding performance of the models. So far, the literature includes only another model trained on this dataset. Kwan Chong Loo [40] proposed the Kranok-NV dataset, and our networks improve on such a model and obtain better results in various metrics. For example, Kwan reports values for Precision, Recall, and F1-score of 99.05%, 97.70%, and 98.37%, respectively, for the normal behavior class. When comparing the results against those obtained by our 3DCNN proposal (Table 3), we obtain better results on the three metrics. For the violent behavior class, a similar situation occurs. Reported results indicate values of 97.44%, 98.94%, and 98.19%. When compared again, the three metrics are outperformed by our 3DCNN proposal (Table 3). Unfortunately, a comparison against other models not trained in this dataset seems unfair since there is a risk of biasing the results to favor our proposal. We plan to compare the performance of our proposal against others from the literature as part of future work.

Data Availability

The Kranok-NV dataset is publicly available at https://www.kaggle.com/kevinbkwanloo/kranoknv.

Conflicts of Interest

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

Acknowledgments

Tecnologico de Monterrey supported this research through the fund for financing the publication of Scientific Articles Initiative.

References

[1] S. E. del Sistema Nacional de Seguridad Pública, *Incidencia delictiva del fuero común, nueva metodología*, GDM, Mexico, 03 2021.

[2] D. G. Kilpatrick, B. E. Saunders, A. Amick-McMullan, C. L. Best, L. J. Veronen, and H. S. Resnick, "Victim and crime factors associated with the development of crime-related post-traumatic stress disorder," *Behavior Therapy*, vol. 20, no. 2, pp. 199–214, 1989.

[3] E. Esbec, "El psicólogo forense en el proceso penal. evaluación psicológica de la víctima," *Psicología forense y tratado jurídicolegal de la discapacidad*, vol. 23, no. 1, pp. 153–217, 2000.

[4] F. M. Donald, "Information processing challenges and research directions in cctv surveillance," *Cognition, Technology & Work*, vol. 21, no. 3, pp. 487–496, 2018.

[5] B. C. Welsh and D. P. Farrington, "Effects of closed-circuit television on crime," *The Annals of the American Academy of Political and Social Science*, vol. 587, no. 1, pp. 110–135, 05 2003.

[6] H. M. Hodgetts, F. Vachon, C. Chamberland, and S. Tremblay, "See no evil: cognitive challenges of security surveillance and monitoring," *Journal of Applied Research in Memory and Cognition*, vol. 6, no. 3, pp. 230–243, 09 2017.

[7] G. A. Martínez-Mascorro, J. R. Abreu-Pederzini, J. C. Ortiz-Bayliss, A. Garcia-Collantes, and H. Terashima-Marín, "Criminal intention detection at early stages of shoplifting cases by using 3d convolutional neural networks," *Computation*, vol. 9, no. 2, p. 24, 2021.

[8] J. Li, X. Jiang, T. Sun, and k. Xu, "Efficient violence detection using 3d convolutional neural networks," *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, in Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–8, IEEE, September 2019.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, Santiago, Chile, December 2015.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[11] P. Singh and V. Pankajakshan, "A deep learning based technique for anomaly detection in surveillance videos," *2018 Twenty Fourth National Conference on Communications (NCC)*, in Proceedings of the 2018 Twenty Fourth National Conference on Communications (NCC), pp. 1–6, IEEE, February 2018.

[12] G. Morales, I. Salazar-Reque, J. Telles, and D. Díaz, "Detecting violent robberies in cctv videos using deep learning, IFIP advances in information and communication technology," in *Artificial Intelligence Applications and Innovations*, J. MacIntyre, I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds., Springer International Publishing, Cham, Switzerland, pp. 282–291, 2019.

[13] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Mathematical Problems in Engineering*, vol. 2013, pp. 1–11, 2013.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, http://arxiv.org/abs/14004.3978.

[15] K. O. Stanley, "Neuroevolution: a different kind of deep learning," *Radar*, vol. 23, p. 334, Jul 2017.

[16] A. Azzini, M. Dragoni, and A. Tettamanzi, "A neuro-evolutionary approach to electrocardiographic signal classification," *Evolution, Complexity and Artificial Life*, vol. 22, pp. 193–207, 2014.

[17] I. Sekaj, L. Cíferský, and M. Hvozdík, "Neuro-evolution of mobile robot controller," *MENDEL*, vol. 25, no. 1, pp. 39–42, 06 2019.

[18] M. Risto, "Evolution of neural networks," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '19*, pp. 694–709, Association for Computing Machinery, New York, NY, USA, July 2019.

[19] C. Dhiman and D. K. Vishwakarma, "A review of state-of-the-art techniques for abnormal human activity recognition," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 21–45, 2019.

[20] N. Li, X. Wu, D. Xu, H. Guo, and W. Feng, "Spatio-temporal context analysis within video volumes for anomalous-event detection and localization," *Neurocomputing*, vol. 155, pp. 309–319, 2015.

[21] M. Manosha Chathuramali, S. Ramasinghe, and R. Rodrigo, "Abnormal activity recognition using spatio-temporal features," *7th International Conference on Information and Automation for Sustainability*, in *Proceedings of the 7th International Conference on Information and Automation for Sustainability*, Colombo, Sri Lanka, December 2014.

[22] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 06 2011.

[23] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, and S. Avidan, "Graph embedded pose clustering for anomaly detection," pp. 10536–10544, 06 2020, http://arxiv.org/abs/1912.11850.

[24] A. A. Chaaraoui, J. R. Padilla-Lopez, and F. Florez-Revuelta, "Abnormal gait detection with RGB-d devices using joint motion history features," *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, in Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), IEEE, May 2015.

[25] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, "Silhouette-based human action recognition using sequences of key poses," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1799–1807, 2013, Smart Approaches for Human Action Recognition.

[26] V. Vignesh, G. Yadav, and A. Sethi, "Abnormal event detection on bmtt-pets 2017 surveillance challenge," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2161–2168, Honolulu, HI, USA, July 2017.

[27] L. Xie and A. Yuille, "Genetic cnn," 2017, http://arxiv.org/abs/1703.01513.

[28] E. Real, S. Moore, A. Selle et al., "Large-scale evolution of image classifiers," 2017, http://arxiz.org/abs/1703.01041.

[29] T. Elsken, J. H. Metzen, and F. Hutter, "Efficient multi-objective neural architecture search via lamarckian evolution," 2019, http://arxiz.org/abs/1804.09081.

[30] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 2, pp. 185–212, 2009.

[31] K. O. Stanley, "Compositional pattern producing networks: a novel abstraction of development," *Genetic Programming and Evolvable Machines*, vol. 8, no. 2, pp. 131–162, June 2007.

[32] S. Risi and K. O. Stanley, "An enhanced hypercube-based encoding for evolving the placement, density, and connectivity of neurons," *Artificial Life*, vol. 18, no. 4, pp. 331–363, 08 2012.

[33] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 54–65, 1994.

[34] J. C. F. Pujol and R. Poli, "Evolving the topology and the weights of neural networks using a dual representation," *Applied Intelligence*, vol. 8, no. 1, pp. 73–84, 01 1998.

[35] D. Whitley and L. Pyeatt, "A comparison between cellular encoding and direct encoding for genetic neural networks," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, pp. 81–89, MIT Press, 1996.

[36] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[37] D. E. Moriarty and R. Miikkulainen, "Efficient reinforcement learning through symbiotic evolution," *Machine Learning*, vol. 22, no. 1–3, pp. 11–32, 1996.

[38] R. Miikkulainen, J. Liang, E. Meyerson et al., "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, R. Kozma, C. Alippi, Y. Choe, and F. C. Morabito, Eds., Elsevier, Amsterdam, 2018.

[39] J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen, "Evolutionary neural automl for deep learning," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2019)*, Prague, July 2019.

[40] K. B. Kwan Chong Loo, *Detection of violent behavior in open environments using pose estimation and neural networks*, PhD thesis, Instituto Tecnologico y de Estudios Superiores de Monterrey, Mexico, 2020.

[41] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA, June 2008.

[42] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "Openpose: realtime multi-person 2d pose estimation using part affinity fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 129–221, 2019.

[43] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," CVPR, 2017, http://arxiv.org/abs/1704.07809.

[44] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," CVPR, 2017, http://arxiv.org/abs/1611.08050.

[45] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," CVPR, 2016, http://arxiv.org/abs/1906.04104.

[46] D. Osokin, "Real-time 2d multi-person pose estimation on cpu: Lightweight openpose," 2018, http://arxiv.org/abs/1811.12004.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6822467, 14 pages
https://doi.org/10.1155/2022/6822467

Research Article

Modeling and Estimation of CO_2 Emissions in China Based on Artificial Intelligence

Pan Wang,[1] **Yangyang Zhong**,[2,3,4] **and Zhenan Yao**[5]

[1]*State Key Laboratory of Nuclear Resources and Environment, East China University of Technology, Nanchang, Jiangxi 330013, China*
[2]*School of Information Engineering, East China University of Technology, Nanchang, Jiangxi 330013, China*
[3]*Jiangxi Engineering Laboratory on Radioactive Geoscience and Big Data Technology, East China University of Technology, Nanchang 330013, Jiangxi, China*
[4]*State Key Laboratory of Palaeobiology and Stratigraphy (Nanjing Institute of Geology and Palaeontology, CAS), Nanjing, Jiangsu 210008, China*
[5]*Engineering Research Center for Seismic Disaster Prevention and Engineering Geological Disaster Detection of Jiangxi Province (East China University of Technology), Nanchang, Jiangxi 330013, China*

Correspondence should be addressed to Yangyang Zhong; zhyy168@ecut.edu.cn

Received 3 March 2022; Accepted 20 June 2022; Published 7 July 2022

Academic Editor: Seyed Jalaleddin Mousavirad

Since China's reform and opening up, the social economy has achieved rapid development, followed by a sharp increase in carbon dioxide (CO_2) emissions. Therefore, at the 75th United Nations General Assembly, China proposed to achieve carbon peaking by 2030 and carbon neutrality by 2060. The research work on advance forecasting of CO_2 emissions is essential to achieve the above-mentioned carbon peaking and carbon neutrality goals in China. In order to achieve accurate prediction of CO_2 emissions, this study establishes a hybrid intelligent algorithm model suitable for CO_2 emissions prediction based on China's CO_2 emissions and related socioeconomic indicator data from 1971 to 2017. The hyperparameters of Least Squares Support Vector Regression (LSSVR) are optimized by the Adaptive Artificial Bee Colony (AABC) algorithm to build a high-performance hybrid intelligence model. The research results show that the hybrid intelligent algorithm model designed in this paper has stronger robustness and accuracy with relative error almost within ±5% in the advance prediction of CO_2 emissions. The modeling scheme proposed in this study can not only provide strong support for the Chinese government and industry departments to formulate policies related to the carbon peaking and carbon neutrality goals, but also can be extended to the research of other socioeconomic-related issues.

1. Introduction

Global warming has become a fact generally accepted by the international community. Climate warming has seriously affected the living environment and social development of humankind. Although the cyclical changes in the natural environment itself affect global climate change, more and more studies have shown that human activities have accelerated the global warming process largely. Since the industrial revolution, with the development of social economy and the increasing intensity of human activities, the concentration of carbon dioxide (CO_2) in the atmosphere has risen sharply [1]. Studies have shown that the main cause of global warming in the past 50 years is the massive emission of greenhouse gases. The main greenhouse gas is CO_2, and its emissions are inextricably linked to the extensive use of fossil fuels and the widespread destruction of forests. [2–5]. The fourth assessment report of IPCC (Intergovernmental Panel on Climate Change) pointed out that China's CO_2 emissions exceeded that of the United States in 2006, ranking first in the world [6]. Moreover, China's CO_2 emissions have continued to increase in recent years. In 2012, the annual emissions reached 8106.43 million tons. Socioeconomic development is necessary to improve living standards and social welfare. To this end, the government must maintain a stable economic growth rate in the

country. Therefore, all countries in the world are facing the dual challenges of CO_2 emissions and social-economic development. On the one hand, it is necessary to curb the level of CO_2 emissions caused by the consumption of fossil fuels, and on the other hand, the economic growth rate needs to be maintained [2, 7]. China's economy has grown exponentially since China implemented its reform and opening-up policy. Studies have shown that China's GDP has grown by an average of more than 9% per year in the past four decades [8–10]. However, this rapid GDP growth has been achieved through a massive increase in energy consumption and the accompanying increase in CO_2 emissions. In recent years, due to the increasingly strong voice in the international community for the development of a low-carbon economy, the issue of CO_2 emissions has attracted widespread attention from scholars around the world, and some research results have been achieved. Scholars' research on CO_2 emissions mainly focuses on two aspects, one is to explore the social and economic factors that affect CO_2 emissions, and the other is to establish CO_2 emissions prediction models.

The factors affecting CO_2 emissions are complex and diverse. In recent years, many experts have done a lot of research work on CO_2 emissions and published some research results. Zha et al. [11] built an assessment framework for direct and indirect CO_2 emissions from the tourist sector and used the SBM-Undesirable model to integrate the CO_2 emission component in the efficiency evaluation framework. Liu et al. [12] examined the dynamic relationship between tourism income, economic growth, energy consumption, and CO_2 emissions. Mujtaba et al. [13] examined the correlation between economic growth, energy consumption, population, trade openness, and CO_2 emissions. By using the generalized method of moments (GMM) dynamic panel model, Al-Ayouty et al. [14] found a positive correlation between population and CO_2 emissions. Aller et al. [15] comprehensively analyzed the factors associated with CO_2 emissions using Bayesian model. The researchers' study was based on data from different countries and the results varied widely [16], but there were some commonalities. Many researchers have found that economic growth and population growth are two of the most important factors that influence CO_2 emissions. Economic growth affects CO_2 emissions mainly through three channels: scale effect, structural effect and technical effect [17, 18]. The scale effect of the economy has a promoting effect on CO_2 emissions, while the structural effect and technological effect have a curb effect on CO_2 emissions [19]. In a study based on provincial panel data, Wang et al. [20] found that economic growth was an important driver of CO_2 emissions growth. Using the generalized index method, Shao et al. [21] found that the scale effect of economy is the main contributor to the increase in CO_2 emissions. Therefore, an increasing number of scholars are currently working to estimate the link between GDP growth and CO_2 emissions [8, 22–26]. Guo et al. [27] found a positive correlation between population size and CO_2 emissions. Ohlan [28] studied the relationship between population density, energy consumption, GDP growth, and CO_2 emissions and found that population density has a strong positive impact on CO_2 emissions, which is the main factor affecting CO_2 emissions. In addition, many other factors can affect CO_2 emissions [29]. Previous research results show that there are many factors affecting CO_2 emissions, and the impact mechanism is relatively complex. The relationship between CO_2 emissions and its influencing factors varies greatly due to regional differences [30].

At present, the research on the impact mechanism of CO_2 emissions is relatively mature. Recently, many researchers have begun to focus on the modeling and prediction of CO_2 emissions issue. Sun et al. [31] have established CO_2 emission prediction models based on BP neural network and least square support vector machine, respectively, and found that the least square support vector machine model has better prediction effect. Budiono et al. [32] used a multiple linear regression model to predict CO_2 emissions from electricity production in Indonesia and the results show that the model predictions are very close to the actual data. Mardani et al. [33] proposed a neuro-fuzzy model to predict CO_2 emissions, and the results show that the model can effectively predict CO_2 emissions. Gao et al. [34] proposed a fractional grey Riccati model (FGRM (1, 1)); by testing 20 data sets from M-competition, it is shown that the proposed model can be used to predict short-term CO_2 emissions. Zhao and Yang [35] tested the dynamic mode decomposition (DMD) for CO_2 emission prediction, which proves that DMD model has ideal effect for short-term carbon dioxide prediction. Wen and Yuan [36] have designed a new hybrid prediction model (RF-DPSO-BP) for CO_2 emission prediction and tested the validity of the model through the panel data test of Chinese commercial sector. It can be seen that machine learning methods have been gradually applied to CO_2 emission modeling and prediction, and existing studies have also confirmed the feasibility and effectiveness of machine learning for CO_2 emission prediction. However, few studies at the current stage consider the hyperparameters optimization problem of intelligent algorithms such as machine learning applied to CO_2 emission modeling and prediction. Therefore, this research will focus on two aspects of China's CO_2 emissions. On the one hand, based on the data of China's CO_2 emissions and socioeconomic indicators, it will analyze the correlation between CO_2 emissions and various socioeconomic indicators. On the other hand, considering that the CO_2 emission modeling data belong to a small sample training set, this study will use the LSSVR to carry out research work. In order to solve the hyperparameter optimization problem of machine learning modeling, the improved artificial bee colony algorithm is mixed with LSSVR to build a CO_2 emission prediction model based on the hybrid intelligent algorithm. The purpose of this hybrid intelligence model is to accurately predict future CO_2 emissions, which can provide reliable theoretical support for Chinese government departments to formulate policies to achieve dual carbon (carbon peaking and carbon neutrality) goals. Finally, this study will give some policy recommendations to the Chinese government.

2. Theory and Methodology

2.1. Methodology of Least Square Support Vector Regression. The basic idea of LSSVR is to use the known sample data to obtain a best fitting function [37] and then input new sample data according to this function to calculate the corresponding output value. Given a training sample set,

$$T = \{(x_1, y_1), \ldots, (x_l, y_l)\} \in (R^n \times Y)^l, \tag{1}$$

where $x_i \in R^n$, $y_i \in Y = R$, $i = 1, 2, \ldots, l$. The sample input is mapped to a high-dimensional space by nonlinear mapping, and the LSSVR function is constructed as follows:

$$y(x) = \omega^{\mathrm{T}} \varphi(x) + b, \tag{2}$$

where ω is weight vector and b is deviation. The solution of (2) can be transformed into an optimization problem, as in

$$\begin{cases} \min\limits_{\omega,b,\xi}, \quad R = \dfrac{1}{2}\|\omega\|^2 + \dfrac{c}{2}\sum\limits_{i=1}^{l}\xi^2, \\ \\ \text{s.t}, \qquad y_i = \omega^{\mathrm{T}}\varphi(x_i) + b + \xi_i, \end{cases} \tag{3}$$

where c is penalty parameter and $\xi_i \geq 0$ is relaxation factor. The function model of LSSVR can be solved by using Lagrange function and KKT optimization condition:

$$y(x) = \sum_{i=1}^{l} a_i K(x, x_i) + b, \tag{4}$$

where a_i is Lagrange multiplier and $K(x, x_i)$ is a kernel function. In this study, the Gauss radial basis function with strong fitting ability is used as the kernel function of LSSVR, as in

$$K(x, x_i) = \exp\left(-\frac{\|x - x_i\|_2^2}{2\sigma^2}\right). \tag{5}$$

2.2. Methodology of Adaptive Artificial Bee Colony (AABC) Algorithm

2.2.1. Artificial Bee Colony Algorithm. As a new biomimetic optimization algorithm, artificial bee colony (ABC) has been widely used in recent years. In the ABC algorithm, a honey source is used to represent a function solution, and the quality of the honey source reflects the merits of the solution.

In the ABC algorithm, the bee colony is composed of a lead bee, a follower bee, and a scout bee, and the problem is solved by information exchange, conversion, and mutual cooperation between the three types of bee colonies. When solving the optimization problem, the spatial point is regarded as the location of the honey source, each point represents a possible solution, and N represents the number of honey sources. The quality of honey source x_i ($i = 1, 2, 3, \ldots, N$) corresponds to the fitness value of the solution. Suppose the dimension of the problem to be solved is D, at the tth iteration, the position of the honey source is $X_i^t = (x_{i1}^t x_{i2}^t \ldots x_{iD}^t)^T$, where t is the current number of iterations, and the maximum number of iterations is T. $x_{id} \in (L_d, U_d)$, and M_d and N_d respectively represent the two boundaries of the search space, $d = 1, 2, 3, \ldots, D$. The ABC algorithm is initialized, and the initial position of the honey source is randomly generated in the search space according to formula (6). Then, the lead bee searches for a new source of honey is according to formula (7).

$$x_{id} = M_d + \text{rand}(0, 1)(N_d - M_d), \tag{6}$$

$$v_{id} = x_{id} + \eta(x_{id} - x_{jd}), \tag{7}$$

where d represents a dimension in the solution space; $j \in \{1, 2, 3, \ldots, N\}$, $j \neq i$; η is a random number between $[-1, 1]$, representing the magnitude of the disturbance.

The fitness of the two honey sources is then evaluated and the retention of x_i or v_i is determined according to the greedy algorithm. The fitness value of the solution is calculated according to (8), where the function value of the solution is represented. The probability that the honey source found by the lead bee is followed is calculated according to (9).

$$\text{fit}_i = \begin{cases} \dfrac{1}{(1 + f_i)}, & f_i \geq 0, \\ \\ 1 + abs(f_i), & f_i < 0, \end{cases} \tag{8}$$

$$P_i = \frac{\text{fit}_i}{\sum_{i=1}^{N} \text{fit}_i}. \tag{9}$$

The follower bee uses roulette to choose to lead bee; that is, a uniformly distributed random number r is generated in $[0, 1]$. When $P_i > r$, the follower bee will generate a new honey source around the honey source and use the greedy algorithm to determine the retained honey source. If a better honey source is not found after t iterations reach the threshold limit, the honey source x_i^t is discarded. At this time, the lead bee becomes a scout bee, a new honey source x_i^{t+1} is randomly generated in the search space instead of x_i^t, and the calculation formula of x_i^t is (10). On the contrary, the honey source x_i^t is retained. At this time, $t = t + 1$, and it is judged whether the algorithm reaches the termination condition and the optimal solution is output.

$$x_i^{t+1} = \begin{cases} L_d + \text{rand}(0, 1)(U_d - L_d), & t \geq \text{limit}, \\ x_i^t, & t < \text{limit}. \end{cases} \tag{10}$$

2.2.2. Improved Artificial Bee Colony Algorithm. Two key problems are involved in the ABC algorithm. One is the selection of the search step size. The search range and search accuracy of the population are controlled by the step size η, but the value of η is completely random. When the value of η is too large, it is easy to jump out of the true global optimal solution; when the value of η is too small, it is easy to converge earlier and obtain a local optimal solution. According to the analysis of bee feeding behavior, there should be a large search step in the early stage of the search,

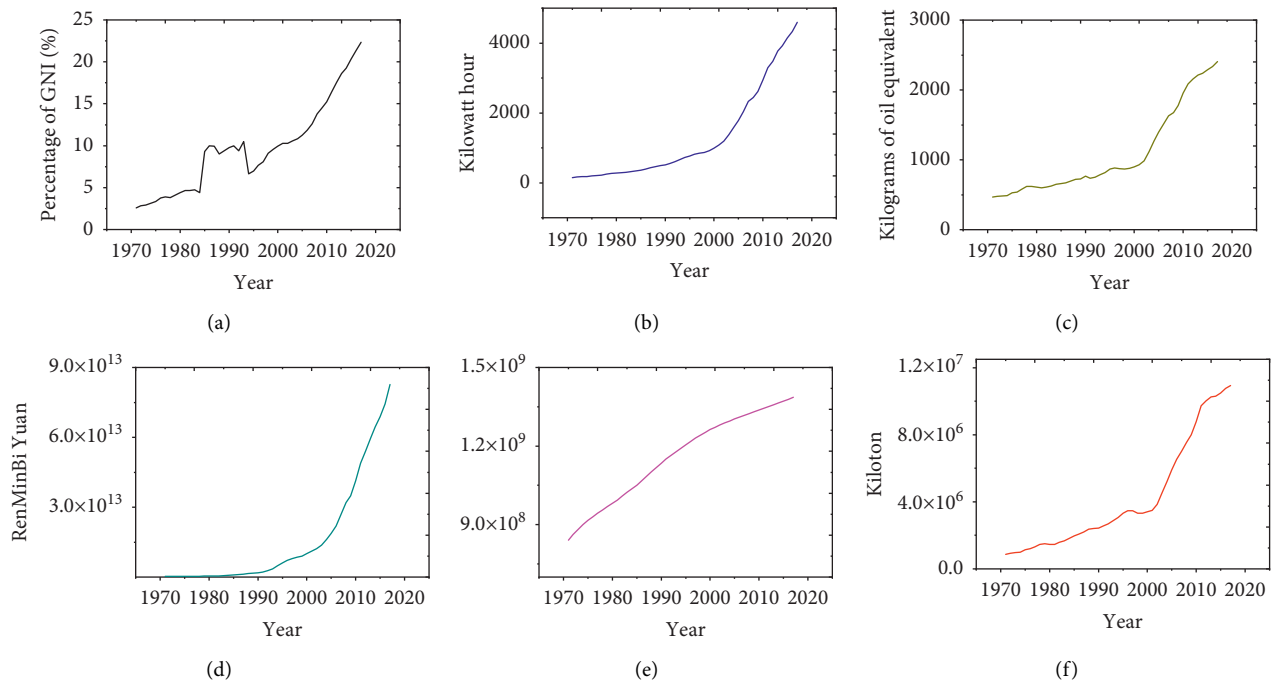


Figure 1: Time-dependent curve of socioeconomic indicators related to CO_2 emissions. (a) FCC. (b) PC per capita. (c) OC per capita. (d) GDP. (e) Population. (f) CO_2 emissions.

so as to quickly gather around the global optimal solution and improve the convergence speed. In the later stage of the search, the optimal solution obtained by the algorithm search is close to the real optimal solution. At this time, the search step size should be gradually reduced to achieve a detailed search around the optimal solution. For this reason, this study designs a self-adaption. The step size of the search strategy is adapted to the decreasing operator, and the calculation formula is

$$\begin{cases} v_{id} = x_{id} + \eta\left(x_{id} - x_{jd}\right), \\ \eta = \eta_0\left(1 - \frac{1}{2}\log_T^t\right), \end{cases} \tag{11}$$

where $\eta_0 \in (0,1)$; $t = \{1,2,3,\ldots,T\}$.

Another key issue is the selection of honey sources in the ABC algorithm. In the standard ABC algorithm, only the honey source with high fitness is selected, and the honey source with low fitness is quickly eliminated. This will lead to a decline in the diversity of the population, which will make it difficult for the algorithm to achieve global optimality. Aiming at this problem, this study designs a new adaptive probability selection mechanism, which is calculated as

$$\begin{cases} P_i = \lambda\frac{\mathrm{fit}_i}{\sum_{i=1}^{N}\mathrm{fit}_i} + (1-\lambda)\frac{1/\mathrm{fit}_i}{\sum_{i=1}^{N}1/\mathrm{fit}_i}, \\ \lambda = e^{(t/T)\ln 2} - 1, \end{cases} \tag{12}$$

where $\lambda \in (0,1)$. In the early stage of the algorithm, $t/T \longrightarrow 0, \lambda \longrightarrow 0$. At this time, the honey source with poor fitness has the possibility of being selected, thereby improving the ability of the algorithm to obtain the generalization; in the later stages of the algorithm's operation, $t/T \longrightarrow 1, \lambda \longrightarrow 1$. At this time, only the honey source with high fitness is selected. Thereby, the convergence speed of the algorithm is accelerated, and the convergence precision of the algorithm is also improved.

In summary, this study designs the adaptive step-down operator to update the step size of the search strategy and designs the adaptive probability selection mechanism to solve the honey source selection problem, improves the ABC algorithm, and establishes an AABC algorithm.

3. Data Preparation and Analysis

3.1. Data Preparation. Database preparation is a very important step for neural network modeling. In this study, the data used were derived from five socioeconomic indicators related to CO_2 emissions in China from the World Development Indicators database published by the World Bank, including fixed capital consumption (FCC), power consumption (PC), oil consumption (OC), gross domestic product (GDP), population (P), and CO_2 emissions. The experimental samples totaled 47 sets of data from 1971 to 2017. Figure 1 is a data view of each indicator. It can be seen that the five socioeconomic indicators related to CO_2 emissions are showing an increasing trend year by year.

Table 1 presents the statistical analysis of the six variables. As shown in Table 1, the value of the standard deviation of CO_2 emissions was found to be equal to 3.2443×10^6. Further, the same statistics for FCC, PC, OC, GDP, and P were found to be equal to 5.302, 1.3254×10^3, 610.2162, 2.3125×10^{13} and 1.6623×10^8 respectively. The

Table 1: Summary of the social and economic indicators related to CO_2 emissions for China.

	FCC (%)	PC (kw·h)	OC (kgoe)	GDP (RMB Yuan)	P	CO_2 emissions (kiloton)
Minimum value	2.5739	151.9893	464.9332	2.4569×10^{11}	8.4111×10^{8}	8.7663×10^{5}
Maximum value	22.3399	4.5916×10^{3}	2.4066×10^{3}	8.2712×10^{13}	1.3864×10^{9}	1.0932×10^{7}
Average	9.6912	1.2906×10^{3}	1.0732×10^{3}	1.6128×10^{13}	1.1599×10^{9}	4.1852×10^{6}
Standard deviation	5.3020	1.3254×10^{3}	610.2162	2.3125×10^{13}	1.6623×10^{8}	3.2443×10^{6}

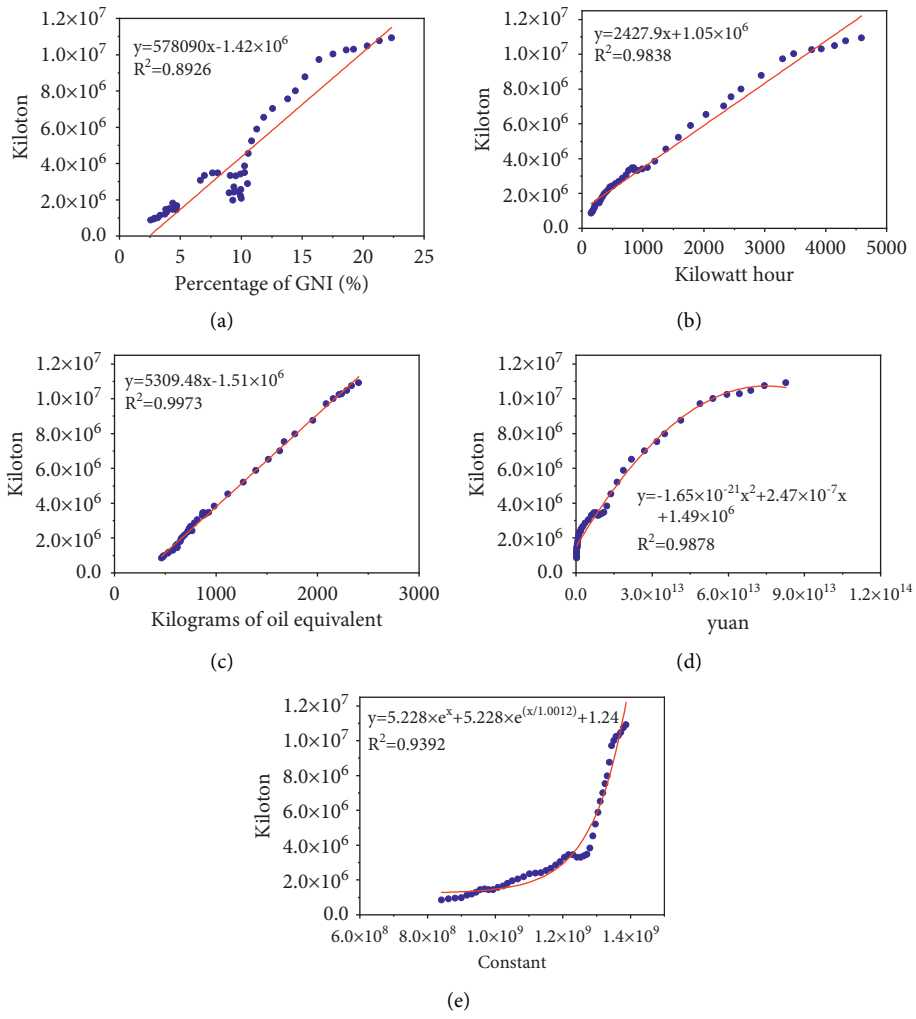


Figure 2: Cross-plots of economic indicators and CO_2 emissions. (a) FCC-CO_2 emissions. (b) PC-CO_2 emissions. (c) OC-CO_2 emissions. (d) GDP-CO_2 emissions. (e) P-CO_2 emissions.

remainder of the results derived from this analysis is set out in Table 1.

3.2. Data Analysis. In order to evaluate the relationship between socioeconomic-related indicators and CO_2 emissions, this study used two analytical methods to characterize the sensitivity of various socioeconomic indicators to CO_2 emissions. A single factor regression analysis was used to obtain a graph of the relationship between each indicator and CO_2 emissions, as shown in Figure 2. In the regression analysis, the coefficient R^2 is used to measure the degree of influence of different indicators on CO_2 emissions, and R^2 represents the proportion of the population variance, which can be obtained by

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (f(x_i) - y_i)}{\sum_{i=1}^{n} f(x_i)^2 - \sum_{i=1}^{n} (y_i)^2 / n}. \quad (13)$$

It can be seen from Figure 2 that there is a positive correlation between each indicator and CO_2 emissions, and R^2 is very large, indicating that there is a good correlation between various economic indicators and CO_2 emissions. Fixed capital consumption, electricity consumption, and oil consumption are all linearly related to CO_2 emissions. However, both GDP and total population are nonlinearly related to CO_2 emissions.

	CO_2 emissions	FCC (%)	PC (kw·h)	OC (kgoe)	GDP (RMB)	P
CO_2 emissions	1	0.9448	0.9918	0.995	0.9666	0.8568
FCC (%)	0.9448	1	0.9398	0.9396	0.9179	0.9849
PC (kw·h)	0.9918	0.9398	1	0.9945	0.989	0.9849
OC (kgoe)	0.995	0.9396	0.9945	1	0.9695	0.985
GDP (RMB)	0.9666	0.9179	0.989	0.9695	1	0.988
P	0.8568	0.9849	0.9849	0.985	0.988	1

Figure 3: Graph of correlation matrix.

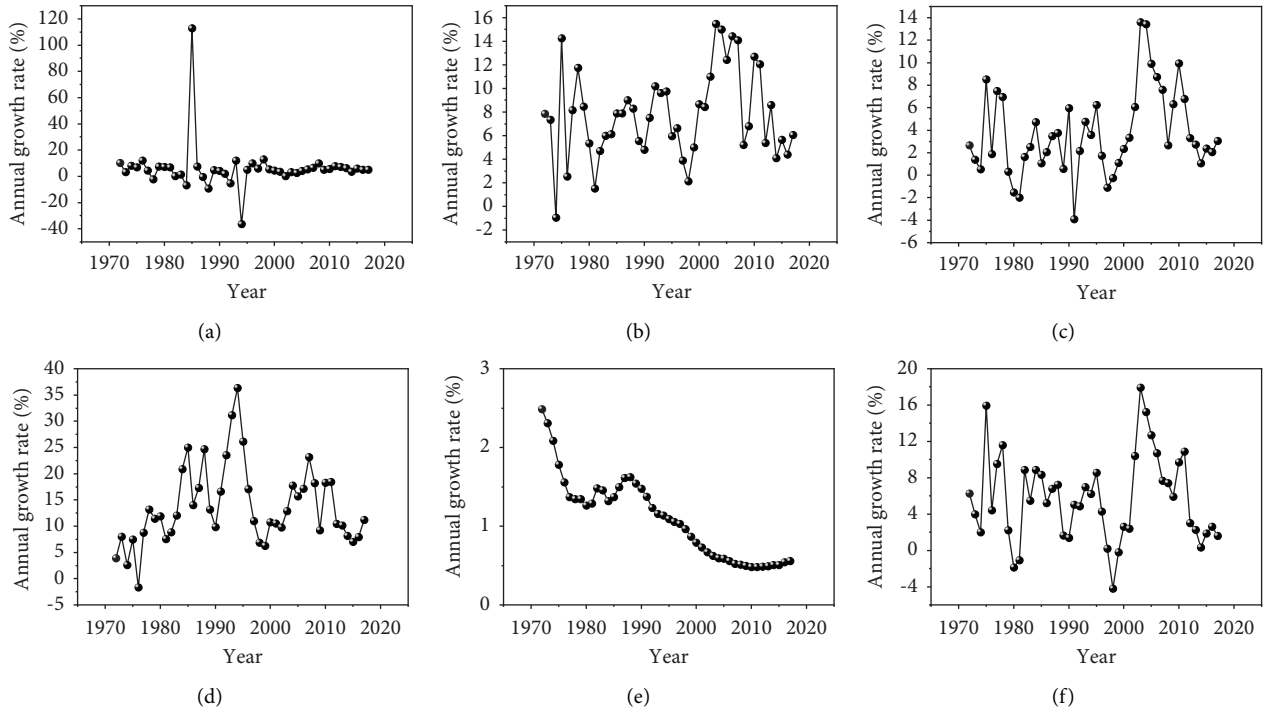


Figure 4: The annual growth rate of socioeconomic indicators and CO_2 emissions. (a) FCC growth rates. (b) PC growth rates. (c) OC growth rates. (d) GDP growth rates. (e) P growth rates. (f) CO_2 emissions growth rates.

In addition to the simple regression method, we also used the correlation coefficient analysis method to analyze the relationship between economic indicators and CO_2 emissions. The analysis results are shown in Figure 3. It can be seen that in addition to the total population indicators, other economic indicators have very high correlation coefficients. Both the crossplot method and the correlation coefficient analysis method show that there is a good correlation between the five socioeconomic indicators selected in this study and the CO_2 emissions, which can be used as the input independent variable of the CO_2 emission prediction model.

In order to reflect the dynamic development of various social and economic indicators and the changes in CO_2 emissions during the study period, we calculated the annual growth rates of various socioeconomic indicators and CO_2 emissions. Figure 4 shows the annual growth rate of socioeconomic indicators and CO_2 emissions. It can be seen from the figure that except for a few years, the annual growth rate of FCC tends to be stable and the fluctuation is small; the annual growth rate of the population shows a trend of decreasing year by year. The annual growth rate of each parameter fluctuates greatly, and the change is relatively complicated.

Table 2 presents the statistical analysis of growth rates for the six variables. As shown in Table 2, the value of the standard deviation of CO_2 emissions growth rate was found to be equal to 4.7735. Further, the same statistics for FCC,

Table 2: Summary of the social and economic indicators related to CO_2 emissions for China.

	FCC	PC	OC	GDP	P	CO_2 emissions
Initial growth rate	10.0202	7.8444	2.6317	3.887	2.4878	6.2675
Average growth rate	5.8882	7.7536	3.7044	13.7197	1.0936	5.7432
Minimum nonnegative growth rate	0.0872	1.4889	−3.9301	0.306	2.5942	0.4803
Maximum growth rate	112.7308	15.4519	13.5682	36.3418	2.4878	17.9247
Standard deviation	17.7822	3.7368	3.7586	7.4978	0.5206	4.7735

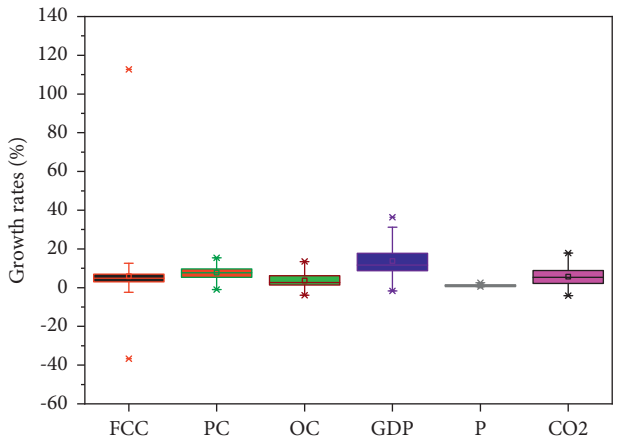


Figure 5: Boxplot showing the differences in values of each variable.

PC, OC, GDP, and P growth rates were found to be equal to 17.7822, 3.7368, 3.7586, 7.4978, and 0.5206, respectively. The remainder of the results derived from this analysis is set out in Table 2. Figure 5 presents the descriptive statistics in the form of a boxplot.

4. Results and Discussion

4.1. Model Establishment. Formulas (7) and (9) in the original ABC algorithm are respectively replaced by the adaptive step size updated formula (11) and the adaptive honey source selection formula (12), composing an improved ABC algorithm called adaptive ABC algorithm. The AABC algorithm is combined with LSSVR to construct an artificial intelligence model for CO_2 emissions prediction. Figure 6 shows the flow of model construction. Considering that the dimensions of each socioeconomic indicator are different and the numerical values show large differences in magnitude, which is not conducive to modeling and calculation, we need to normalize the data first. Set the parameters of AABC, including the number of bee colonies NP, the maximum number of searches for honey sources limit, the maximum number of iterations T of the algorithm, the number of parameters to be optimized D, and the range of values of each parameter. Then, set the objective function. In this study, the minimum root mean square error (RMSE) fitted by LSSVR is taken as the objective function of the AABC algorithm, and its reciprocal is defined as the fitness function. Finally, through the iterative optimization of AABC algorithm, the optimal parameters of LSSVR are obtained; that is, the AABC-LSSVR model is successfully

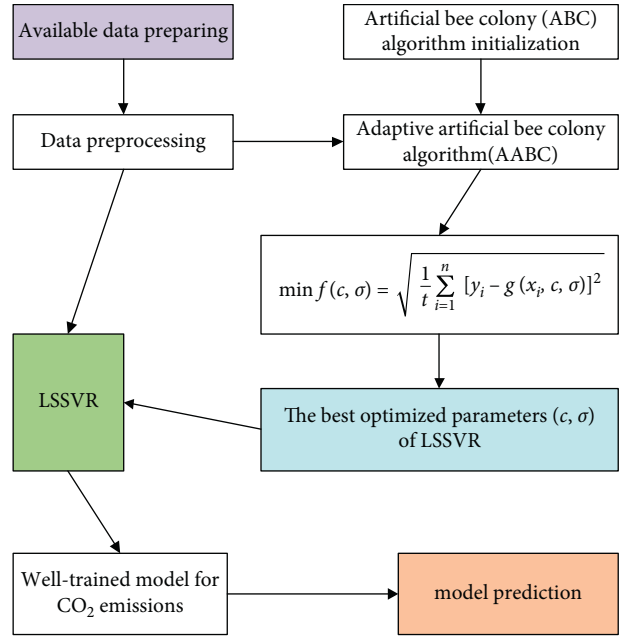


Figure 6: Flow chart of modeling for CO_2 emissions prediction.

constructed, which can be used for the prediction of CO_2 emissions.

4.2. Model Performance. As mentioned above, the data in this study are the socioeconomic and CO_2 emissions data from 1971 to 2017 in China, totaling 47 groups. The 47 groups of data are divided into three parts: training, validation, and testing, of which 70% are trained. The validation and testing sections each accounted for 15%.

When initializing the parameters of the ABC and AABC optimization algorithms, the size of the artificial bee colony is 50, the number of honey sources is half of the size of the bee colony, the dimension D of the solution problem is 2, the maximum number of iterations T is 300, and the maximum number of searches is limit 100. The penalty parameter c and the kernel function parameter σ have values ranging from [0.05, 1000] to [0.0001, 50], respectively. According to the aforementioned modeling process, the program was coded with MATLAB R2018b, which can be used to obtain the LSSVR and AABC-LSSVR models, respectively. Figure 7 shows the optimization results of the AABC algorithm. The optimum c and σ were synchronously obtained as follows: best c = 2.8284, best σ = 0.0625.

Figure 8 is an iterative optimization curve for the AABC-LSSVR model. It can be seen from the figure that under the

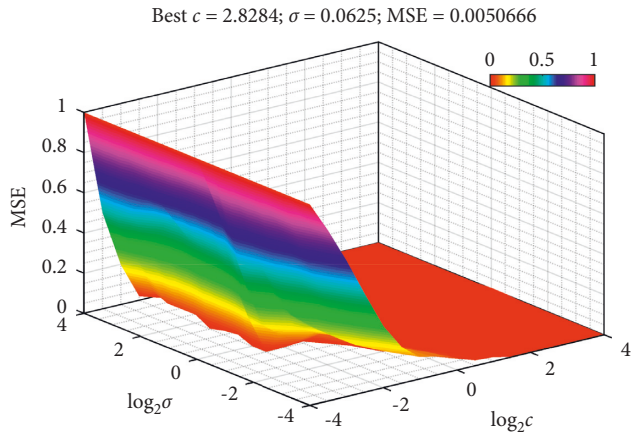


Figure 7: Parameter optimization results of LSSVR by using AABC algorithm.

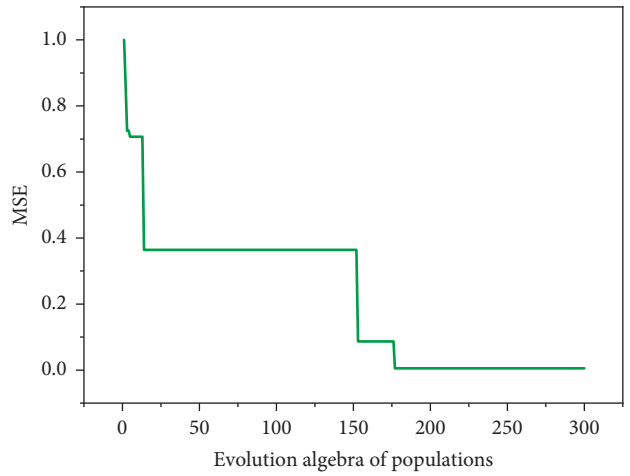


Figure 8: Iterative optimization curve of AABC-LSSVR model.

control of adaptive regulation, in the early stage, the iterative curve drops rapidly (MSE value decreases rapidly), indicating that the iterative step size of the population is larger and the convergence speed is faster; in the medium term, the iterative curve is stable. The state indicates that some honey sources with low fitness are retained, so that the diversity of the population is supplemented; in the later stage, the iterative curve drops rapidly again to reach the convergence value, indicating that only the honey source with high fitness is selected at this time. It can be seen that the AABC has strong adaptability and effectiveness in searching for key parameters of the LSSVR algorithm.

The predictive performance of the LSSVR and AABC-LSSVR models with the same inputs is evaluated using the coefficient of determination (R^2), as shown in Figures 9 and 10. It can be seen from Figures 9(a) and 10(a) that LSSVR has a strong fitting ability in the training set part, which is slightly inferior to the AABC-LSSVR model. However, as can be seen from Figures 9(b), 9(c) and 10(b), 10(c), the LSSVR model is significantly less effective in the verification and testing part than the AABC-LSSVR model, especially in the test section. It shows that the optimization effect of AABC on LSSVR is very obvious, which improves the fitting performance of LSSVR. It also shows that the AABC-LSSVR model proposed in this study has strong generalization ability in the prediction of CO_2 emissions and is trustworthy.

Figures 9(d) and 10(d) show crossplot of real CO_2 emissions versus predicted values. Coefficient of determination (R^2) for LSSVR prediction is equal to 0.9427 which verifies robustness of LSSVR model. While coefficient of determination (R^2) for AABC-LSSVR prediction is equal to 0.997 which verifies strong robustness of improved LSSVR model. It is sufficient to claim that the robustness of LSSVR was improved by using AABC algorithm.

In addition to coefficient of determination (R^2), error distribution was also employed for assessment of the LSSVR and AABC-LSSVR performance. Figure 11 allows more statistical analysis of LSSVR and AABC-LSSVR performance by using error distribution information. As shown in Figure 11(a), mean and standard deviation of error distribution for LSSVR model are in turn equal to 2.1152×10^5 and 2.1152×10^5, which are relatively larger than that of AABC-LSSVR model (see Figure 11(b)). It means that the prediction error of AABC-LSSVR model is more acceptable.

Figure 12 provides an opportunity to compare relative error between LSSVR and AABC-LSSVR for all data points. From Figure 12(a), it can be seen that relative error of AABC-LSSVR for almost all data points is located in the range of [−5% 5%], which is an acceptable value. Figure 12(b) presents the descriptive statistics in the form of a boxplot for relative error data. Small relative error of mean and fluctuation range of data reveals high performance of AABC-LSSVR model in CO_2 emissions prediction.

In summary, we can see that AABC-LSSVR outperformed LSSVR owing to higher R^2 and lower relative error. This study states AABC algorithm is very effective for LSSVR and AABC-LSSVR predicted values are in good agreement with reality.

4.3. Discussion. The model of CO_2 emission prediction has been constructed based on AABC and LSSVR algorithms, and the performance of the model has been tested. The results also show that the AABC-LSSVR model has strong robustness and generalization ability, and the prediction accuracy is acceptable and trustworthy. Therefore, the AABC-LSSVR model will be used to predict China's CO_2 emissions for the next five years (2018–2022). Considering that the socioeconomic indicators for the next five years are also unknown, this study will reduce the uncertainty of prediction by setting different scenarios, and these scenarios will adopt different input independent variable data. The characteristics of China's social and economic indicators have been analyzed in the past few years, and the annual growth rate of each indicator has been calculated. Therefore, four scenarios will be set to generate model input independent variable data, as shown in Table 3. These data are all calculated by (14). Scenario (a) is based on the growth rate calculated in 2017 as the compound annual growth rate. Scenario (b) is based on the average growth rate between 1971 and 2017 as the

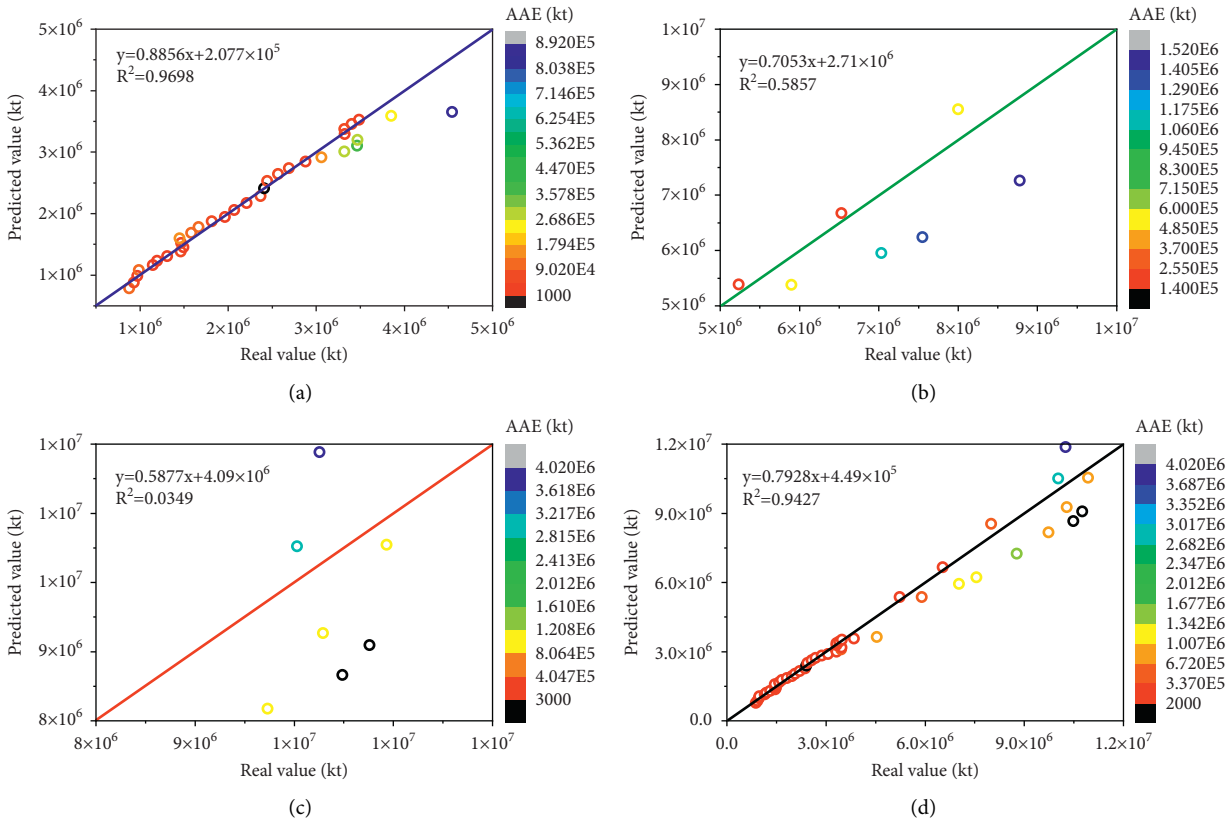


Figure 9: Comparison of the CO_2 emissions prediction results with LSSVR model and real CO_2 emissions. (a) LSSVR—training results. (b) LSSVR—validation results. (c) LSSVR—testing results. (d) LSSVR—all results.

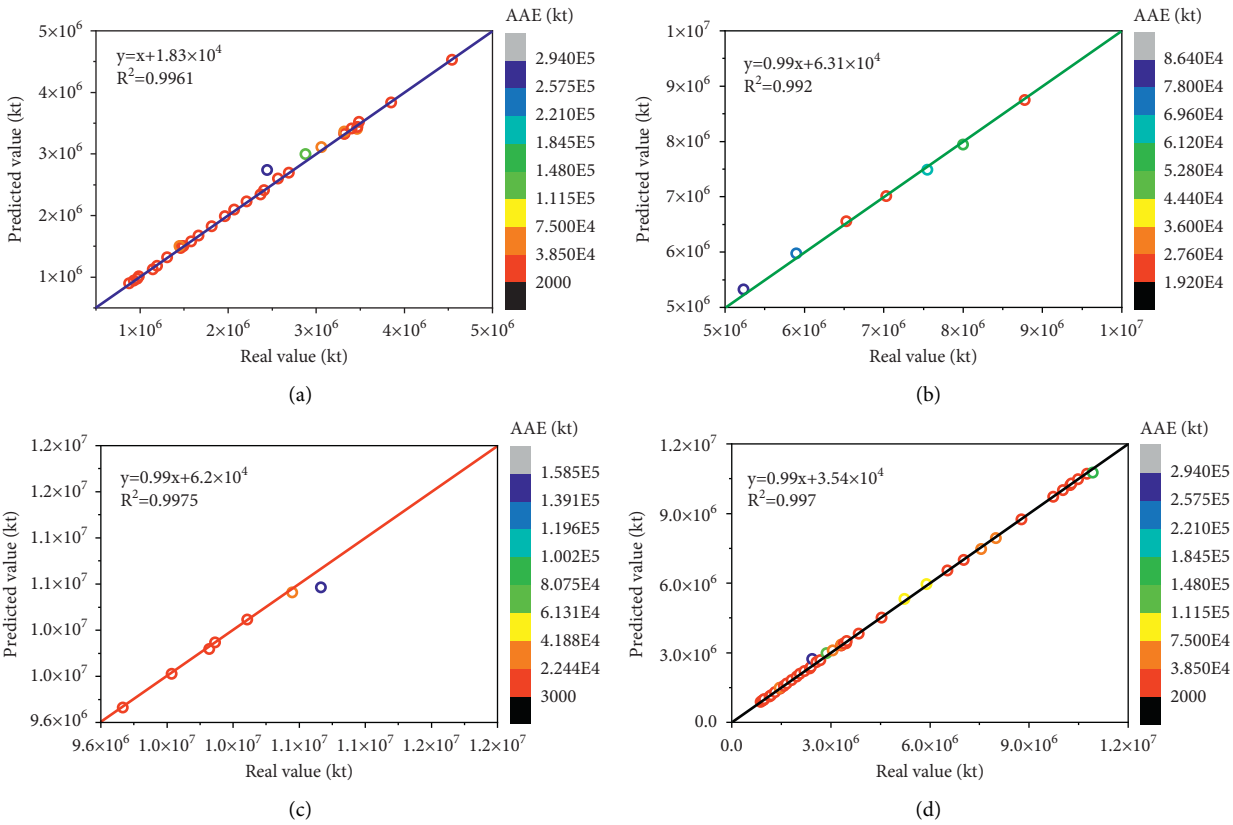


Figure 10: Comparison of the CO_2 emissions prediction results with AABC-LSSVR model and real CO_2 emissions. (a) AABC-LSSVR—training results. (b) AABC-LSSVR—validation results. (c) AABC-LSSVR—testing results. (d) AABC-LSSVR—all results.

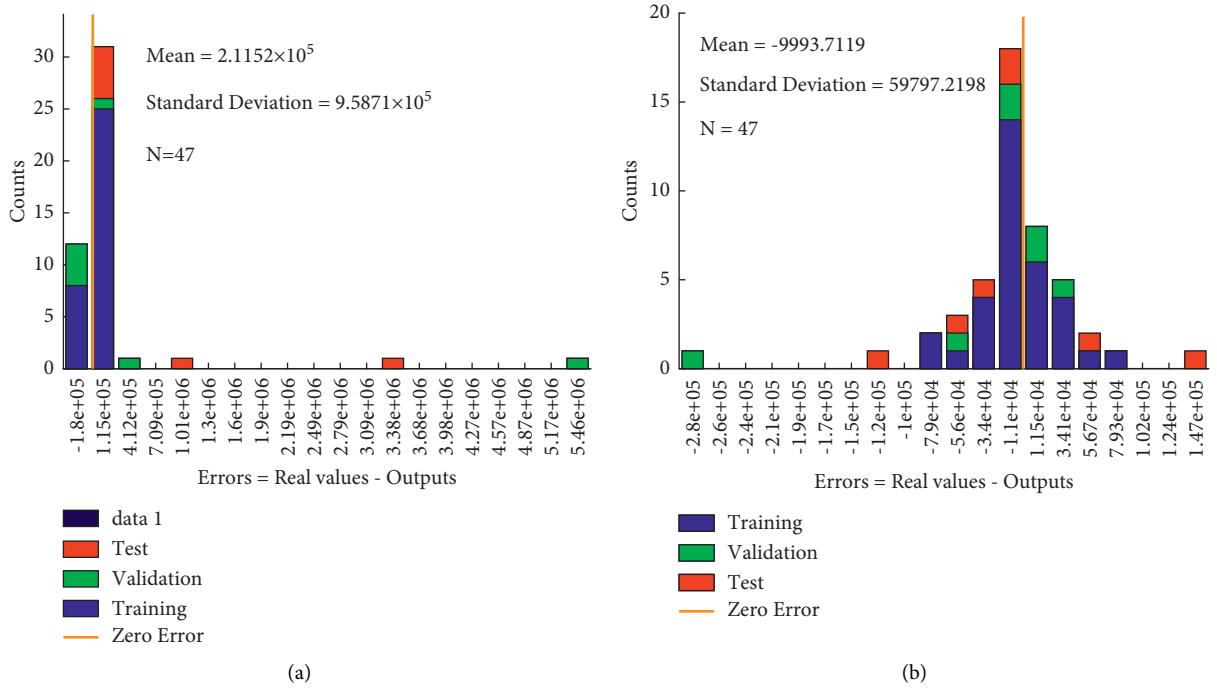


Figure 11: Absolute error histogram with 20 bins of each model. (a) LSSVR model. (b) AABC-LSSVR model.

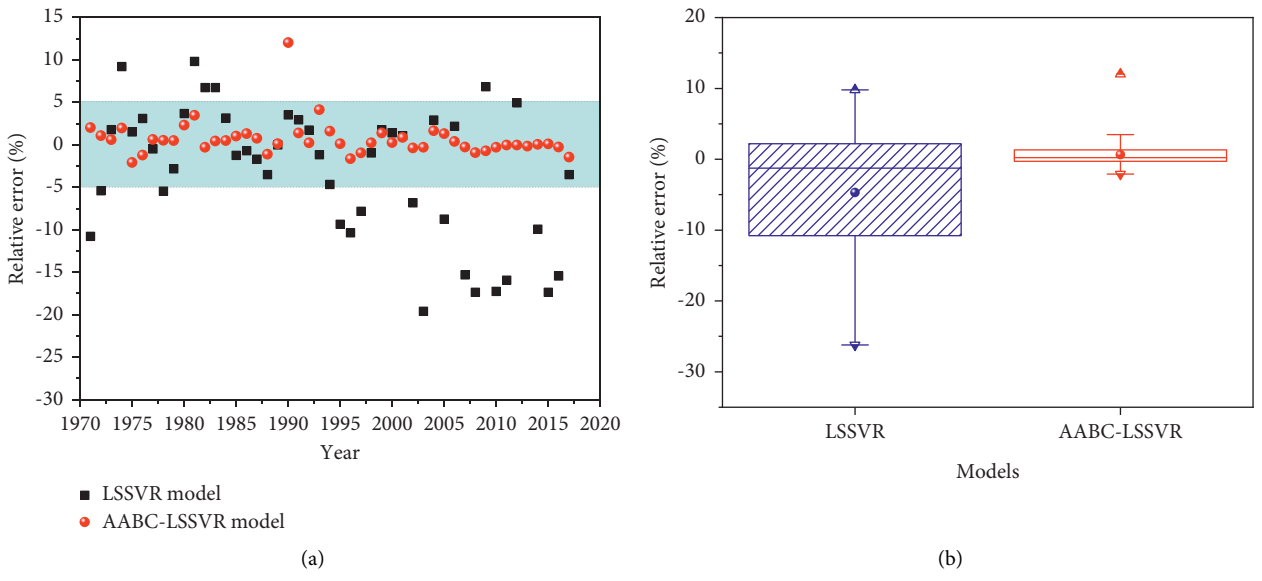


Figure 12: Results of relative error analysis for each model. (a) Relative error of models in prediction of CO_2 emissions. (b) Boxplot showing the differences in relative error of each model.

compound annual growth rate. Scenario (c) is based on the nonnegative minimum growth rate between 1971 and 2017 as the compound annual growth rate. Scenario (d) is based on the nonnegative maximum growth rate between 1971 and 2017 as the compound annual growth rate.

$$f(t_n) = f(t_0)(1 + cagr)^n, \tag{14}$$

where a is the value after n years and b is the initial value.

By substituting the five indicators of CO_2 emissions in each of the scenarios in Table 3 into the AABC-LSSVR model, the predictions of CO_2 emissions in four different scenarios can be obtained, as shown in Figure 13.

As can be seen from Figure 13(a), China's CO_2 emissions in the next five years (2018–2022) have shown an upward trend year by year in the above four scenarios. In scenario (a), CO_2 emissions increase from 1.0932×10^7 kilotons in 2017 to 1.3026×10^7 kilotons in 2022. In scenario (b), CO_2

Table 3: The forecasting scenarios of the social and economic indicators related to CO_2 emissions for China (from 2018 to 2022).

Scenario	Year	FCC (%)	PC (kw·h)	OC (kgoe)	GDP (yuan)	P
Scenario (a)	2018	24.5784	4.5917×10^3	2.4699×10^3	8.5927×10^{13}	1.4209×10^9
	2019	27.0412	5.3402×10^3	2.5349×10^3	8.9267×10^{13}	1.4562×10^9
	2020	29.7507	5.7591×10^3	2.6017×10^3	9.2737×10^{13}	1.4925×10^9
	2021	32.7318	6.2108×10^3	2.6701×10^3	9.6342×10^{13}	1.5296×10^9
	2022	36.0117	6.698×10^3	2.7404×10^3	1.0008×10^{14}	1.5676×10^9
Scenario (b)	2018	23.6613	4.9553×10^3	2.4978×10^3	9.4060×10^{13}	1.4016×10^9
	2019	25.0608	5.3478×10^3	2.5925×10^3	1.0697×10^{14}	1.4169×10^9
	2020	26.5432	5.7714×10^3	2.6908×10^3	1.2164×10^{14}	1.4324×10^9
	2021	28.1132	6.2286×10^3	2.7928×10^3	1.3833×10^{14}	1.4481×10^9
	2022	29.7762	6.7219×10^3	2.8986×10^3	1.5731×10^{14}	1.4639×10^9
Scenario (c)	2018	22.3593	4.6599×10^3	2.4139×10^3	8.4858×10^{13}	1.3931×10^9
	2019	22.3788	4.7293×10^3	2.4214×10^3	8.7059×10^{13}	1.3997×10^9
	2020	22.3983	4.7997×10^3	2.4288×10^3	8.9318×10^{13}	1.4065×10^9
	2021	22.4179	4.8712×10^3	2.4362×10^3	8.1635×10^{13}	1.4132×10^9
	2022	22.4375	4.9437×10^3	2.4437×10^3	8.4012×10^{13}	1.4201×10^9
Scenario (d)	2018	47.5238	5.301×10^3	2.7332×10^3	1.1277×10^{14}	1.4209×10^9
	2019	101.0978	6.1202×10^3	3.1039×10^3	1.5375×10^{14}	1.4562×10^9
	2020	215.0663	7.0658×10^3	3.5251×10^3	2.0963×10^{14}	1.4925×10^9
	2021	457.5122	8.1576×10^3	4.0035×10^3	2.8582×10^{14}	1.5296×10^9
	2022	973.2694	9.4181×10^3	4.5466×10^3	3.8968×10^{14}	1.5676×10^9

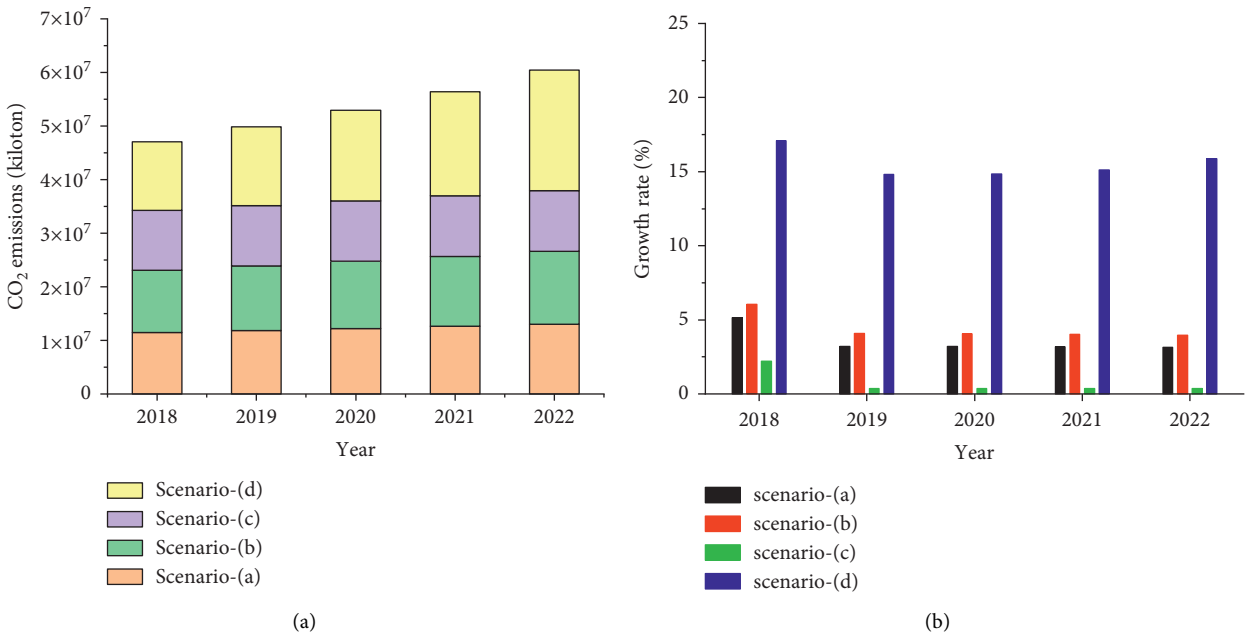


Figure 13: The forecast results of CO_2 emissions for China in the next five years (2018–2022). (a) Prediction values of CO_2 emissions in four scenarios. (b) The growth rates of CO_2 emissions.

emissions increase from 1.0932×10^7 kilotons in 2017 to 1.3575×10^7 kilotons in 2022. The CO_2 emissions in scenario (c) increase from 1.0932×10^7 kilotons in 2017 to 1.1338×10^7 kilotons in 2022. The CO_2 emissions in scenario (d) increase from 1.0932×10^7 kilotons in 2017 to 2.2509×10^7 kilotons in 2022. It can be seen from Figure 13(b) that the annual growth rate of China's CO_2 emissions in the next five years under the four scenarios is quite different. Scenario (a) has an average annual growth rate of 3.57%; scenario (b) has an average annual growth rate of 4.43%; scenario (c) has an average annual growth rate of 0.735%; scenario (d) has an average annual growth rate of 15.54%. In summary, the forecast of China's CO_2 emissions in the next five years under the four scenarios indicates that China's CO_2 emissions in 2022 will fluctuate between 1.1338×10^7 kilotons and 2.2509×10^7 kilotons.

The analysis results of the scenarios set above show that future CO_2 emissions are closely related to their influencing

factors, and changes in any key influencing factor will cause strong fluctuations in CO_2 emissions. Referring to the social and economic development of China in the past period, prediction of CO_2 emissions in the next few years using the hybrid intelligence model is proposed in this study. The results show that China's CO_2 emissions will continue to rise in the future. The Chinese government should formulate special policies for relevant social-economic indicators to effectively control future CO_2 emissions and complete the carbon peaking and carbon neutrality goals as soon as possible.

5. Conclusions

Based on China's CO_2 emissions from 1971 to 2017 and related socioeconomic indicators, including FCC, electricity consumption, OC, GDP, and total population, a new hybrid intelligent algorithm model for China's CO_2 emissions forecast is proposed in this research. In view of the shortcomings of the standard ABC algorithm, in this study, an adaptive step search formula and an adaptive probability selection formula are designed to enhance the optimization ability of ABC, thus forming the AABC algorithm to optimize the hyperparameters of the LSSVR model. Finally, the ABC-LSSVR and AABC-LSSVR models are established respectively to predict China's CO_2 emissions. The results show that the absolute value of the relative error of the AABC-LSSVR model is almost controlled within 5%, which can confidently indicate that the AABC-LSSVR model has better accuracy and generalization ability for CO_2 emissions prediction. It can be seen that the hyperparameter optimization problem of machine learning can be effectively solved by establishing a hybrid intelligent algorithm, thereby greatly improving the reliability of the prediction results in the research of CO_2 emissions advance forecast. According to the forecast results of China's CO_2 emissions from 2018 to 2022 by the AABC-LSSVR model, China's CO_2 emissions will continue to grow in the future for a long time and will fluctuate between 1.1338×10^7 kilotons and 2.2509×10^7 kilotons by 2022. If the Chinese government needs to achieve the dual carbon goal as soon as possible, it is necessary to introduce targeted policies to restrict the social and economic indicators closely related to CO_2 emissions, so as to effectively curb CO_2 emissions in the future.

We make the following policy recommendations to the Chinese government based on the conclusions of the analysis above. First, in order to use fixed capital scientifically and rationally and reduce its physical loss as much as possible, it is necessary to improve the management awareness and management level of enterprises. Second, it is necessary to optimize the energy structure and increase the substitution ratio of new energy to traditional energy. China's energy structure is dominated by petrochemical energy, and a large amount of CO_2 is released in the process of energy consumption. Therefore, increasing the proportion of clean energy such as solar energy and wind energy in the energy consumption structure can effectively reduce CO_2 emissions, thereby alleviating the conflict between CO_2 emissions and economic development. Third, China's industrial structure and export trade structure need to be adjusted. Manufacturing, which has long been China's main export sector, is highly dependent on upstream sectors, leading to more energy consumption and CO_2 emissions. The modern service industry relies more on human capital investment, and its asset-light characteristics can greatly reduce the energy consumption of upstream and downstream industrial chains, which can reduce CO_2 emissions from the source. Therefore, the transformation of the industrial structure from manufacturing to modern service industry can not only improve economic efficiency, but also reduce environmental costs. Fourth, China still needs to vigorously develop its economy to increase residents' income. There is a clear relationship between people's consumption concept and behavior and economic level; that is, the higher the social affluence is, the more likely people will pay attention to the environment and consider whether their behavior will have an impact on the environment. People in high-income societies will obviously reduce "high-carbon" consumption and are more inclined to buy low-carbon products to promote CO_2 emission reduction. Fifth, according to the share of CO_2 emissions, the government should focus on controlling high-emission sectors, such as electricity, heat, gas, and water production and supply industries, and encourage innovation and promotion of energy-saving technologies in these sectors. In addition, since technological progress is an effective way to reduce CO_2 emissions, government departments should strongly support enterprises and scientific research institutions to develop low-carbon emissions, CO_2 capture, and storage technology research and development. Sixth, China's urbanization is in an accelerated stage, and the government should vigorously promote a population structure that is conducive to sustainable development, to effectively control CO_2 emissions. Human activities are the main reason for the increase of CO_2 emissions. The government needs to guide people to green travel and low-carbon life, so that people can actively contribute to CO_2 emission reduction through their own behaviors and achieve the purpose of effectively controlling the impact of population factors on CO_2 emissions.

In addition, machine learning has become a key technology in CO_2 emission forecasting research. Although the hybrid intelligence model proposed in this study has achieved good results, it still has the possibility of further optimization. CO_2 emissions and related influencing factors are time series data. On the one hand, researchers can try to extract features from the original data to the maximum extent for algorithm modeling through feature engineering. On the other hand, researchers can try to expand the deep learning framework of the recurrent neural network series to this study. In short, only continuous in-depth research in the future can continuously improve the accuracy of advanced forecasting of CO_2 emissions.

Data Availability

The data used to support the findings of the 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

The authors would like to thank Xiaoyan Deng for assisting in preparation of this manuscript, particularly grateful to her for checking grammar, style, and syntax. This research was supported by the Scientific Research Foundation of East China University of Technology (Grant no. DHBK2019298), State Key Laboratory of Palaeobiology and Stratigraphy (Nanjing Institute of Geology and Palaeontology, CAS) (no. 203125), and Scientific Research Foundation of East China University of Technology (Grant no. DHBK2019305).

References

[1] U. K. Pata and A. E. Caglar, "Investigating the EKC hypothesis with renewable energy consumption, human capital, globalization and trade openness for China: evidence from augmented ARDL approach with a structural break," *Energy*, vol. 216, Article ID 119220, 2021.

[2] C. L. Fang, S. J. Wang, and G. D. Li, "Changing urban forms and carbon dioxide emissions in China: a case study of 30 provincial capital cities," *Applied Energy*, vol. 158, pp. 519–531, 2015.

[3] Y. Shan, J. Liu, Z. Liu et al., "New provincial CO2 emission inventories in China based on apparent energy consumption data and updated emission factors," *Applied Energy*, vol. 184, pp. 742–750, 2016.

[4] Y. Shan, D. Guan, J. Liu et al., "Methodology and applications of city level CO 2 emission accounts in China," *Journal of Cleaner Production*, vol. 161, pp. 1215–1225, 2017.

[5] S. J. Wang, X. P. Liu, C. S. Zhou, J. C. Hu, and J. P. Ou, "Examining the impacts of socioeconomic factors, urban form, and transportation networks on CO2 emissions in China's megacities," *Applied Energy*, vol. 185, pp. 189–200, 2017.

[6] IPCC, *IPCC guidelines for national greenhouse gas inventories. Intergovernmental Panel on Climate Change*, IPCC, London, 2006.

[7] S. J. Wang, C. L. Fang, Y. Wang, Y. B. Huang, and H. T. Ma, "Quantifying the relationship between urban development intensity and carbon dioxide emissions using a panel data analysis," *Ecological Indicators*, vol. 49, pp. 121–131, 2015.

[8] S. J. Wang, C. L. Fang, X. L. Guan, B. Pang, and H. T. Ma, "Urbanisation, energy consumption, and carbon dioxide emissions in China: a panel data analysis of China's provinces," *Applied Energy*, vol. 136, pp. 738–749, 2014.

[9] D. Guan, Y. Shan, Z. Liu, and K. He, "Performance assessment and outlook of China's emission-trading scheme," *Engineering*, vol. 2, no. 4, pp. 398–401, 2016.

[10] S. J. Wang and X. P. Liu, "China's city-level energy-related CO 2 emissions: s," *Applied Energy*, vol. 200, no. 15, pp. 204–214, 2017.

[11] J. Zha, L. He, Y. Liu, and Y. Shao, "Evaluation on development efficiency of low-carbon tourism economy: a case study of Hubei Province, China," *Socio-Economic Planning Sciences*, vol. 66, pp. 47–57, 2019.

[12] Y. Liu, T. Kumail, W. Ali, and F. Sadiq, "The dynamic relationship between CO_2 emission, international tourism and energy consumption in Pakistan: a cointegration approach," *Tourism Review*, vol. 74, 2019.

[13] A. Mujtaba, P. K. Jena, and D. Mukhopadhyay, "Determinants of CO2 emissions in upper middle-income group countries: an empirical investigation," *Environmental Science and Pollution Research*, vol. 27, no. 30, pp. 37745–37759, 2020.

[14] I. Al-Ayouty, H. Hassaballa, and M. El Hini, "Carbon dioxide emissions: a dynamic panel analysis of energy intensive industries in Egypt," *European Journal of Sustainable Development*, vol. 10, no. 3, p. 109, 2021.

[15] C. Aller, L. Ductor, and D. Grechyna, "Robust determinants of CO_2 emissions," *Energy Economics*, vol. 96, Article ID 105154, 2021.

[16] U. K. Pata and A. Kumar, "The influence of hydropower and coal consumption on greenhouse gas emissions: a comparison between China and India," *Water*, vol. 13, no. 10, p. 1387, 2021.

[17] P. Sheng, J. Li, M. Zhai, and M. U. Majeed, "Economic growth efficiency and carbon reduction efficiency in China: coupling or decoupling," *Energy Reports*, vol. 7, pp. 289–299, 2021.

[18] M. Murshed, B. Saboori, M. Madaleno, H. Wang, and B. Doğan, "Exploring the nexuses between nuclear energy, renewable energy, and carbon dioxide emissions: the role of economic complexity in the G7 countries," *Renewable Energy*, vol. 190, pp. 664–674, 2022.

[19] P. K. Adom, W. Bekoe, F. Amuakwa-Mensah, J. T. Mensah, and E. Botchway, "Carbon dioxide emissions, economic growth, industrial structure, and technical efficiency: empirical evidence from Ghana, Senegal, and Morocco on the causal dynamics," *Energy*, vol. 47, no. 1, pp. 314–325, 2012.

[20] S. J. Wang, C. L. Fang, and Y. Wang, "Spatiotemporal variations of energy-related CO 2 emissions in China and its influencing factors: an empirical analysis based on provincial panel data," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 505–515, 2016.

[21] S. Shao, J. H. Liu, Y. Geng, Z. Miao, and Y. C. Yang, "Uncovering driving factors of carbon emissions from China's mining sector," *Applied Energy*, vol. 166, pp. 220–238, 2016.

[22] U. Al-mulali, J. Y. M. Lee, A. H. Hakim Mohammed, and L. Sheau-Ting, "Examining the link between energy consumption, carbon dioxide emission, and economic growth in Latin America and the Caribbean," *Renewable and Sustainable Energy Reviews*, vol. 26, pp. 42–48, 2013.

[23] M. Sebri and O. Ben-Salha, "On the causal dynamics between economic growth, renewable energy consumption, CO 2 emissions and trade openness: f," *Renewable and Sustainable Energy Reviews*, vol. 39, pp. 14–23, 2014.

[24] A. Kasman and Y. S. Duman, "CO2 emissions, economic growth, energy consumption, trade and urbanization in new EU member and candidate countries: a panel data analysis," *Economic Modelling*, vol. 44, pp. 97–103, 2015.

[25] S. Chaabouni and K. Saidi, "The dynamic links between carbon dioxide (CO2) emissions, health spending and GDP growth: a case study for 51 countries," *Environmental Research*, vol. 158, pp. 137–144, 2017.

[26] M. Mohsin, S. Naseem, M. Sarfraz, and T. Azam, "Assessing the effects of fuel energy consumption, foreign direct investment and GDP on CO2 emission: new data science evidence from Europe & Central Asia," *Fuel*, vol. 314, Article ID 123098, 2022.

[27] W. Guo, T. Sun, and H. J. Dai, "Effect of population structure change on carbon emission in China," *Sustainability*, vol. 8, no. 3, p. 225, 2015.

[28] R. Ohlan, "The impact of population density, energy consumption, economic growth and trade openness on CO2 emissions in India," *Natural Hazards*, vol. 79, no. 2, pp. 1409–1428, 2015.

[29] P. Wang, W. Wu, B. Zhu, and Y. Wei, "Examining the impact factors of energy-related CO2 emissions using the STIRPAT model in Guangdong Province, China," *Applied Energy*, vol. 106, pp. 65–71, 2013.

[30] U. K. Pata and C. Isik, "Determinants of the load capacity factor in China: a novel dynamic ARDL approach for ecological footprint accounting," *Resources Policy*, vol. 74, Article ID 102313, 2021.

[31] W. Sun, H. Jin, and X. Wang, "Predicting and analyzing CO2 emissions based on an improved least squares support vector machine," *Polish Journal of Environmental Studies*, vol. 28, no. 6, pp. 4391–4401, 2019.

[32] R. Budiono, H. Juahir, M. Mamat, S. Supian, and M. Nurzaman, "Modelling the contribution of CO_2 emissions from fuel used on total CO_2 emissions at power generation in Indonesia," *IOP Conference Series: earth and Environmental Science*, vol. 311, no. No. 1, Article ID 012079, 2019.

[33] A. Mardani, Y. Van Fan, M. Nilashi et al., "A two-stage methodology based on ensemble Adaptive Neuro-Fuzzy Inference System to predict carbon dioxide emissions," *Journal of Cleaner Production*, vol. 231, pp. 446–461, 2019.

[34] M. Gao, H. Yang, Q. Xiao, and M. Goh, "A novel fractional grey Riccati model for carbon emission prediction," *Journal of Cleaner Production*, vol. 282, Article ID 124471, 2021.

[35] B. Zhao and W. Yang, "Short-run forecast and reduction mechanism of CO2 emissions: a Chinese province-level study," *Environmental Science and Pollution Research*, vol. 29, no. 9, pp. 12777–12796, 2020.

[36] L. Wen and X. Yuan, "Forecasting CO2 emissions in Chinas commercial department, through BP neural network based on random forest and PSO," *Science of the Total Environment*, vol. 718, Article ID 137194, 2020.

[37] P. Wang and S. P. Peng, "A new scheme to improve the performance of artificial intelligence techniques for estimating total organic carbon from well logs," *Energies*, vol. 11, no. 4, p. 747, 2018.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2340856, 9 pages
https://doi.org/10.1155/2022/2340856

Research Article

The intelligent Traffic Management System for Emergency Medical Service Station Location and Allocation of Ambulances

Ezzatollah Asgharizadeh,[1] **Mahsa Kadivar**,[2] **Mohammad Noroozi**,[3] **Vahid Mottaghi**,[4] **Hamed Mohammadi**,[3] **and Adel Pourghader Chobar**[5]

[1]*Department of Industrial Management, University of Tehran, Tehran, Iran*
[2]*Department of Business Management, Payame Noor University, Varamin Branch, Tehran, Iran*
[3]*Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran, Iran*
[4]*Department of IT Management, Qeshm Branch, Islamic Azad University, Qeshm, Iran*
[5]*Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

Correspondence should be addressed to Ezzatollah Asgharizadeh; asghari@ut.ac.ir

Received 22 April 2022; Revised 10 June 2022; Accepted 16 June 2022; Published 7 July 2022

Academic Editor: Diego Oliva

In the present study, the optimization of medical services considering the role of intelligent traffic management is of concern. In this regard, a two-objective mathematical model of a medical emergency system is assessed in order to determine the location of emergency stations and determine the required number of ambulances to be allocated to the station. The objective functions are the maximization of covering the emergency demands and minimization of total costs. Moreover, the use of an intelligent traffic management system to speed up the ambulance is addressed. In this regard, the proposed two-objective mathematical model has been formulated, and a robust counterpart formulation under uncertainty is applied. In the proposed method, the values of the objective function increase as the problem becomes wider and, with a slight difference in large dimensions, converge in terms of the solution. The numerical results indicate that, as the problem's complexity increases, the robust optimization method is still effective because, with the increasing complexity of the problem, it can still solve large-scale problems in a reasonable time. Moreover, the difference between the value of the objective function in the proposed method and the presence of uncertainty parameters is very small and, in large dimensions, is quite logical and negligible. The sensitivity analysis shows that, with increasing demand, both the number of ambulances required and the amount of objective function have increased significantly.

1. Introduction

Emergency medical services are an important part of the health system, an effective system that provides rapid response to calls, patient transfer, and timely treatment, ultimately, saving human lives, accountable to the prehospital health system [1]. Medical emergency organizations are divided into two main categories, affiliated to the United Kingdom and the United States and affiliated to France and Germany, including on-site treatment, and the accelerated response has not been significant [2].

On the other hand, in healthcare management systems, time is a significant and influential factor. In large cities, however, there are always significant delays in the arrival of patients' positions due to heavy traffic. For this reason, it is essential to design a traffic management system in order to provide required routes for health vehicles and minimize the travel time to the patient's position [1, 3].

Accordingly, this research is based on the use of uncertainty to respond to healthcare visiting demand as quickly as possible. In this regard, stochastic models target the inherent uncertainty and dynamics of the medical emergency system and are adapted to take into account the randomness of the initial contact process. Therefore, location-inspired models also consider demand-constrained costs and try to minimize the total cost of setting up stations and meeting demand.

To address demand uncertainty, first a definitive model and then a probabilistic model are presented, and it is assumed that each vehicle is capable of delivering a maximum certain number of emergency calls over the planning horizon. Moreover, like most coverage models, they need a vehicle that can meet emergency demand [4, 5]. The initial formulation is then expanded, and a two-step mathematical program is proposed to combine the uncertainties so that this is the first attempt to apply this method to an ambulance location, where two decision steps are considered. The first step is to determine the location of the standby locations and the second step is to assign the task of entering service requests to the standby locations. It should be noted that demand uncertainty manifests itself in the second stage. Next, an ambulance allocation model was proposed. In that model, minimizing the costs of operating and transporting the ambulance, as well as the requests that were not submitted on time, was addressed. To offer the best possible solution taking into account the diversity of demand, the newly introduced model provides a strong example of location-allocation formulation that aims to ensure that a solution to a set of demands can be solved. In a way, the concept shows the maximum number of simultaneous requests to estimate the number of vehicles to be stationed at each station.

In previous research, location optimization to meet emergency needs has been highly regarded. Goli and Malmir [6] have located and routed in crisis situations. In this research, the covering approach has been used. In this method, not all places with demand are serviced, and an attempt is made to cover the demand of the affected people in the nearest places. Fuzzy uncertainty has been used in this research. Pahlevan et al. [7] have examined the location and distribution of goods in conditions of uncertainty. In this research, gray wolf optimizer (GWO) and red deer algorithm (RDA) have been applied to solve the problem.

Zhang and Li [8] have proposed a mathematical model for locating emergency relief centers. In this research, fuzzy uncertainty and possibility planning have been used. Nickel et al. [9] have studied the problem of locating ambulance stations in the event of random demand. For this purpose, a sampling method is developed, and then, by optimizing a mathematical model, the best position of the stations is determined. Boujemaa et al. [10] have proposed a two-stage model for medical service in which service centers and service time are optimized.

After careful consideration of the previous research items, it is revealed that the uncertainty of demand has attracted the attention of researchers, and, according to their objectives and the field under study, the allocation of emergency vehicles has been considered in many different ways. However, it is challenging to compare their results at a common point. In addition, the robust formulation is still a considerable challenge given the computational effort required to solve them. One of the most useful data analysis tools is robust optimization suitable for decision-making under uncertainty. An intelligent traffic management system has also been used to accelerate the movement of vehicles and reach the patient as quickly as possible. In other words, the contribution of this research can be summarized as the following items:

(i) Integrating the home healthcare planning and intelligent traffic management

(ii) considering two-stage programming to manage the intelligent traffic system and home healthcare routing

(iii) considering uncertainty in demand parameters for home healthcare optimization and providing robust counterpart formulation to deal with such uncertainty

In this study, a robust formation is presented to formulate the problem with demand uncertainty. This formulation ensures with a certain probability that the number of vehicles stationed at a location can meet the maximum number of simultaneous demands. The sampling approach has been implemented to select the ambulance standby locations as well as the number of ambulances in each of the selected locations. In this case, it is assumed that demand follows a certain discrete distribution. Next, the proposed approach seeks to minimize fixed costs while ensuring the level of coverage for all scenarios considered. The results, based on a set of random cases, confirm the use of a random approach to deal with the problem under study. In a relatively similar work, a two-stage stochastic location model is proposed for the design of a robust two-stage medical emergency system, which simultaneously measures the location of ambulance stations, the number, and type of ambulances that must be deployed and determines the required areas of each station. Next, the average sample approximation algorithm is used to solve the problem.

Therefore, in this research, an attempt has been made to use more up-to-date methods in solving optimization problems and, in particular, robust optimization, which solves the problems caused by uncertainties in the input data and in the model parameters, which are discussed in Section 2. Next, the findings are analyzed in Section 3, and in Section 4, conclusions are provided.

2. Problem Description

When an emergency call is made in a subarea, if there is one, the ambulance is transported from the station to the scene of the accident, and then the patient is transported to the hospital. Therefore, the sooner the patient is taken to the hospital, the sooner the treatment begins. In general, ambulances are located in preinstalled stations. When a call is made, the ambulance leaves the station at the scene and then returns to the station. The allocation of ambulances to the place of demand is shown in Figure 1.

Moreover, in this research, the intelligent traffic management system has been considered using a map. It predefines the ambulance route and the traffic light on the route and the status of the light (green or red) on the map. In this system, there is a control center that detects the current location of the ambulance and sends a command to the traffic light to turn the light green, or the light will continue

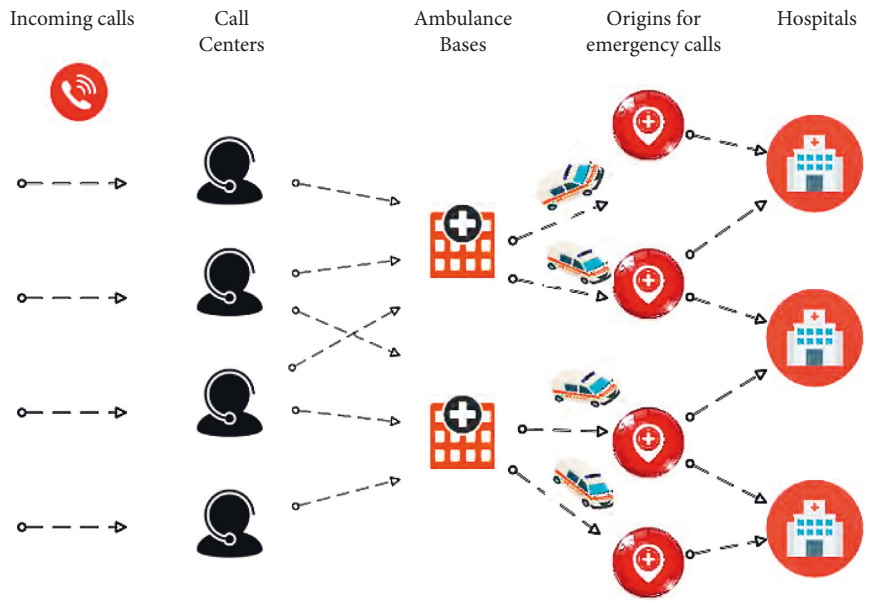


Figure 1: Overview of demand and allocation of ambulances to the demand place.

to its current state. The goal is for the ambulance not to get in its way due to a red light.

The mechanism of this system is that it calculates the distance and direction of the ambulance, i.e., the distance between the ambulance and the next traffic light, and calculates the direction, i.e., determining the ambulance route at each intersection. Calculating the distance allows the control center to turn the traffic light green at the right time so that there is no traffic before the intersection, and calculating the route allows the control center to identify the traffic light at the next intersection. When the ambulance starts moving, its initial position is sent to the control center to determine the direction of movement and then to identify the nearest traffic light on the route. If the distance from the ambulance to the traffic light is less than a fixed value, the traffic light will show green; otherwise, the light will continue to its current state. It is the responsibility of radio systems to detect the movement of an ambulance, and a route is adopted that transports the ambulance to the scene of the accident or the place of request or the patient from the place of request to the hospital in the shortest possible time. After identifying this optimal route, the lights in the route and the distance of the ambulance to each intersection or lights are determined by the location and special radio systems to make arrangements at the desired time to minimize oncoming traffic.

The notations used in the research, including the sets, parameters, and decision variables, are expressed as follows.

2.1. Sets and Parameters

I: set of demand points

J: set of potential locations for stations

L: traffic light potential location set

R: set of ambulances that leave the station

f_j: cost (daily) of building at station j

p_j: cost of maintaining and purchasing each ambulance at station j

d_{ij}: distance between the place of demand i and station j

C: cost per shipping unit

μ_i: average demand (daily) at the place of demand i

q_i: maximum simultaneous number of demands at the place of demand i

β: standard distance for applying the intelligent traffic management system l

H_l : cost (daily) of each unit of the intelligent traffic management system l

M: a very large number

W: the weight of unmet demand

N_l: the number of ambulances that, according to each demand, can be covered by the potential location of the intelligent traffic management system l

2.2. Decision Variables

X_{ij}: percentage of demand at location i covered by station j

Y_j: a binary variable and equal to 1 if the station is constructed at potential location j; otherwise, it is 0

N_j: number of ambulances available at station j

W_{rl}: a binary variable and equal to 1 when the distance from ambulance r to light l is greater than β; otherwise, it is 0

Z_{rl}: a binary variable and equal to 1 when in light l for ambulance r turns green; 0 is the state when the light turns red

S_l: a binary variable and equal to 1 when the traffic light l is in the potential location; otherwise, it is 0

2.3. Objective Function and Constraints. The certain model used to minimize the total cost, based on previous research [11, 12], is as follows:

$$P: \min \sum_{j\in J} f_j Y_j + \sum_{j\in j} P_j N_j + \sum_{j\in J}\sum_{i\in J} cd_{ij}\mu_i X_{ij} + \sum_{l} H_l S_l, \quad (1)$$

$$\sum_{j\in J}^{s.t.} X_{ij} = 1, \forall i \in I, \quad (2)$$

$$X_{ij} \le Y_j, \forall i \in I, \forall j \in J, \quad (3)$$

$$N_j \le MY_j, \forall j \in J, \quad (4)$$

$$N_l \le MS_l, \forall l \in L, \quad (5)$$

$$W_{rl} + Z_{rl} = 1, \forall l \in L, \forall r \in R, \quad (6)$$

$$\sum_{i\in I} q_i X_{ij} \le N_j, \forall j \in J, \quad (7)$$

$$0 \le X_{ij} \le 1, \forall i \in I, \forall j \in J, \quad (8)$$

$$Y_j \in \{0,1\}, Z_{rl} \in \{0,1\}, W_{rl} \in \{0,1\}, S_l \in \{0,1\}, \forall j \in J, \quad \forall l \in L, \forall r \in R, \quad (9)$$

$$N_j \in Z^+, N_l \in Z^+ \forall j \in J, \forall l \in L. \quad (10)$$

The objective function in (1) includes minimizing the cost of construction of medical emergency system stations, the cost of emergency transportation equipment, the average cost of transportation between medical emergency system stations and the place of demand, and, finally, the cost of setting up and equipping an intelligent traffic management system. Constraint (2) guarantees the processing of each demand. Constraints (3) and (4) indicate the allocation of demand and transportation equipment only to open stations. Constraint (5) indicates the allocation of the intelligent traffic management system to the lights. Constraint (6) states how the intelligent traffic system operates when the distance from ambulance r to lamp l is less than the defined value. Constraint (7) states that the number of ambulances at station j should not be less than the maximum simultaneous demand. Constraint (8) represents the Xij range, and constraints (9) and (10) also represent the binary nonnegative variables.

In order to estimate the response rate of the medical emergency system, a maximum coverage model has been used, which ensures that the requests answered by EMS stations have a transfer time less than T [13, 14]. This concept is equivalent to minimizing the demands that are not covered at the time [15, 16]. Therefore, the following model is introduced to minimize fixed costs, equipment costs and purchases, and the cost of fines for applications that were not covered in the previous model.

$$O: \min \sum_{j\in J} f_j Y_j + \sum_{j\in j} P_j N_j + W \sum_{j\in J}\sum_{i\in I_j} \mu_i X_{ij} + \sum_{l} H_l S_l, \quad \text{s.t.} \quad \text{Constraint}(2) \sim \text{Constraint}(10). \quad (11)$$

Here, W indicates the weight of applications that were not completed within the specified time. Moreover, in this objective function, I_j is defined as $I/\{i: d_{ij} \le T \times V\}$ where V is the average ambulance speed. By combining the two-objective functions (1) and (11), a definite two-objective model is created to design the problem of the medical emergency system, which will be used to compare with the robust model.

2.4. Robust Counterpart Formulation. In most cases of designing a medical emergency system, sufficient research data is not available to achieve a possible distribution function of parameters that include uncertainty. Therefore, developing a robust peer-to-peer approach is the most likely solution. In this research, the definite model used has become its robust counterpart by replacing the constraints that include the definite parameter with the constraints that include the set of uncertainties. That is, the uncertainty in the two parameters of the number of requests and the maximum simultaneous demand is considered. Moreover, the set of elliptical uncertainties along with safe parameters has been used to describe the uncertainties in this research.

In order to implement a robust counterpart model, first by inspiring the foundation of robust optimization, especially in the research of Ben-Tal et al. [17] and Goodarzian et al. [18], a standard formulation is created to convert the model to a robust counterpart form. The research is then applied to the mathematical model. Moreover, in order to solve the robust model, the parameters related to the robust counterpart form are quantified, and then the mathematical model is optimized.

2.5. Robust Formulation of Model P. To formulate a robust counterpart of the Model P, it is assumed that the number of emergency calls belongs to a set of elliptical uncertainties, which is shown in

$$U = \left\{U \in \mathbb{R}^{|I|}: U = \overline{U} + \Delta\varsigma, \varsigma \le \theta, \overline{U} \in \mathbb{R}^{|I|}, \Delta \in \mathbb{R}^{|I|\times|I|}, \theta \in \mathbb{R}\right\}, \quad (12)$$

where U is the vector of uncertain demand at places of demand and its numerical values are

$$U = [\mu_1, \mu_2, \ldots, \mu_n]^T, \quad n = |I|, \quad \overline{U} = [\overline{\mu}_1, \overline{\mu}_2, \ldots, \overline{\mu}_n]^T. \quad (13)$$

The matrix $\triangle = \sum^{1/2}$ can be obtained through a separation method. Σ is actually the covariance matrix of emergency calls. θ is the safe parameter that is determined by

derivatives and is also selected by the decision-maker to show the appropriate reaction to his approach to risk.

Moreover, the following constraints are considered for introducing the auxiliary variable t_j.

$$t_j \geq \sum_{i \in I} d_{ij} \mu_i X_{ij}, \forall j \in J. \tag{14}$$

Accordingly, the robust counterpart approach can be obtained through the following model:

$$\max_{\mu \in U} \sum_{i \in I} d_{ij} \mu_i X_{ij}, \tag{15}$$

$$\max_{\varsigma \leq \theta} (\overline{U} + \Delta\varsigma)^T X_j \leq t_j, \tag{16}$$

where $X_j = [d_{1j}X_{1j} + d_{2j}X_{2j} + \ldots + d_{nj}X_{nj}]^T, n = |I|$. Therefore, based on [15], it can be concluded that the robust counterpart of Constraint (7) can be reformulated as

$$\overline{U}^T X_j + \theta \sqrt{X_j^T \sum X_j} \leq t_j \forall j \in J. \tag{17}$$

This means hedging against the standard distribution θ, provided that the coefficients belong to a set of elliptical uncertainties. The MNCD-based cone uncertainty set will be

$$\mathbb{Q} = \left\{ \mathcal{Q} \in \mathbb{R}^{|I|} : \mathcal{Q} = \overline{\mathcal{Q}} + \Xi\xi, \xi \leq \beta, \overline{\mathcal{Q}} \in \mathbb{Q}^{|I|}, \Xi \in \mathbb{R}^{|I| \times |I|}, \beta \in \mathbb{R} \right\}, \tag{18}$$

where numerical values are $\overline{\mathcal{Q}} = [\overline{q}_1, \overline{q}_2, \ldots, \overline{q}_n]$ and $\mathcal{Q} = [q_1, q_2, \ldots, q_n]$. Moreover, Ψ is the covariance matrix, and β is the safe parameter. Therefore, the robust counterpart of Constraint (7) will be as formulated as

$$\overline{\mathcal{Q}}^T X_j + \beta \sqrt{X_j^T \Psi X_j \leq N}. \tag{19}$$

As a result, the robust counterpart of the Model P will be as

$$P\text{: } \min \sum_{j \in J} f_j Y_j + \sum_{j \in j} P_j N_j + \sum_{j \in J} ct_j + \sum_l H_l S_l, \tag{20}$$

$$\sum_{i \in I}^{s.t.} d_{ij} \overline{\mu}_i X_{ij} + \theta \sqrt{\sum_{i \in I} \sigma_i d_{ij}^2 X_{ij}^2} \leq t_j, \forall j \in J, \tag{21}$$

$$\sum_{i \in I} \overline{q}_i X_{ij} + \beta \sqrt{\sum_{i \in I} \Psi_{ij}^2 X_{ij}^2} \leq N_j, \forall j \in J, \tag{22}$$

$$X_{ij} \in R_+ \forall i \in I, j \in J. \tag{23}$$

2.6. Robust Formulation of Model O. The robust formulation of Model O is formulated in a similar way to Model P. The third part in the objective function (11) is related to the uncertainty in the number of emergency calls that belong to the elliptical set, which is shown in

$$V_j = \left\{ V \in \mathbb{R}^{|I_j|} : V = \overline{V} + \Lambda\varsigma, \varsigma \leq \gamma, \overline{V} \in \mathbb{R}^{|I_j|}, \Lambda \in \mathbb{R}^{I_j \times I_j}, \gamma \in \mathbb{R} \right\}, \tag{24}$$

where V is the uncertain emergency demand vector generated by the following demand location $V, i \in I_j, V = [\mu_{i_1}, \mu_{i_2}, \ldots, \mu_{i_m}]^T, m = |I_j|$. Moreover, $\Lambda = \phi^{1/2} \cdot \phi$ is the covariance matrix; μ and γ are safe parameters. Similar to the method used in Constraint (14), we formulate a robust constraint counterpart as

$$\widetilde{t}_j \geq \sum_{i \in I_j} \mu_i X_{ij}, \forall j \in J, \tag{25}$$

$$\overline{V}^T X_j + \gamma \sqrt{X_j^T \phi X_j} \leq \widetilde{t}_j, \forall j \in J. \tag{26}$$

Therefore, the robust counterpart of Model O can be formulated as

$$O\text{: } \min \sum_{j \in J} f_j Y_j + \sum_{j \in j} P_j N_j + W \sum_{j \in J} \widetilde{t}_j + \sum_l H_l S_l, \tag{27}$$

$$\sum_{j \in J}^{s.t,} X_{ij} = 1, \forall i \in I, \tag{28}$$

$$X_{ij} \leq Y_j, \forall i \in I, \forall j \in J, \tag{29}$$

$$N_l \leq MS_l, \forall l \in L, \tag{30}$$

$$W_{rl} + Z_{rl} = 1, \forall l \in L, \forall r \in R, \tag{31}$$

$$\sum_{i \in I} \overline{q}_i X_{ij} + \beta \sqrt{\sum_{i \in I} \Psi_i X_{ij}^2} \leq N_j, \forall j \in J, \tag{32}$$

$$\sum_{i \in I} \mu_i X_{ij} + \gamma \sqrt{\sum_{i \in I_j} \sigma_i X_{ij}^2} \leq \widetilde{t}_j, \forall j \in J, \tag{33}$$

$$0 \leq X_{ij} \leq 1, \forall i \in I, \forall j \in J, \tag{34}$$

$$Y_j \in \{0, 1\}, Z_{rl} \in \{0, 1\}, W_{rl} \in \{0, 1\}, S_l \in \{0, 1\}, s \quad \forall j \in J, \forall l \in L, \forall r \in R. \tag{35}$$

Finally, a summary of the medical emergency system problem formulas is provided by considering the intelligent traffic management system as a robust two-objective optimization problem and is presented in

$$\min \sum_{j \in J} f_j Y_j + \sum_{j \in j} P_j N_j + \sum_{j \in J} ct_j + \sum_l H_l S_l, \tag{36}$$

$$\min \sum_{j \in J} f_j Y_j + \sum_{j \in j} P_j N_j + W \sum_{j \in J} \widetilde{t}_j + \sum_l H_l S_l, \tag{37}$$

$$\sum_{j \in J}^{s.t.} X_{ij} = 1, \forall i \in I, \tag{38}$$

$$X_{ij} \leq Y_j, \forall i \in I, \forall j \in J, \tag{39}$$

$$N_l \leq MS_l, \forall l \in L, \tag{40}$$

$$W_{rl} + Z_{rl} = 1, \forall l \in L, \forall r \in R, \tag{41}$$

$$\sum_{i\in I} d_{ij}\overline{\mu}_i X_{ij} + \theta\sqrt{\sum_{i\in I}\sigma_i d_{ij}^2 X_{ij}^2} \le t_j, \forall j \in J, \tag{42}$$

$$\sum_{i\in I}\overline{q}_i X_{ij} + \beta\sqrt{\sum_{i\in I}\Psi_i X_{ij}^2} \le N_j, \forall j \in J, \tag{43}$$

$$\sum_{i\in I}\mu_i X_{ij} + \gamma\sqrt{\sum_{i\in I_j}\sigma_i X_{ij}^2} \le \widetilde{t}_j, \forall j \in J, \tag{44}$$

$$0 \le X_{ij} \le 1, \forall i \in I, \forall j \in J, \tag{45}$$

$$0 \le X_{ij} \le 1, \forall i \in I, \forall j \in J, \tag{46}$$

$$N_j \in Z_+, \forall j \in J, \tag{47}$$

$$t_j, \widetilde{t}_j \in R_+, \forall j \in Js, \tag{48}$$

3. Numerical Results

In this section, in line with the previous sections and the process of achieving the research results, numerical tests are used to model the medical emergency system by considering the intelligent traffic management system. In this series of tests, the location of the emergency service stations and the facilities that are actually located in them are determined from the candidate stations for medical emergency services. Then, the required number of ambulances is determined based on the share of demand and its amount in each station. The data used in this section is based on the study of Ndiaye and Alfares [19], in which the location of the stations is proportional to the demand, which is dependent on the conditions in summer and winter. In order to better understand, the data used for solving the models are described in the following subsection.

3.1. Test Problem Design. The proposed model is an integer quadratic problem that can be solved through the bifurcation and delimitation algorithm to obtain the optimal solution [20–23]. Both certain and robust models have been solved using GAMS software and Windows 10 using a laptop that is equipped with an Intel Core i7 processor 8 GB of RAM. First, the performance of the proposed model is examined by comparing the two-objective functions that have been stated, namely, cost and effect functions.

A small case study has been conducted considering three emergency stations and five places of demand. The operating cost of allocating vehicles at stations $j = 1, 2, 3$ is 11,000,000, 56,000, and 130,000 units, respectively. The cost of each shipping unit is 50. The maximum simultaneous demand for the places of demand $i = 1, 2, 3, 4, 5$ is considered: 90, 19, 39, 183, and 103, respectively. The cost of the uncovered application is 50. The cost of station equipment for stations $j = 1, 2, 3$ is 115,700, 125,000, and 270,000 units, respectively.

Table 1: The values of $\overline{\mu}$.

i	$\overline{\mu}_i^l$	$\overline{\mu}_i^u$	$\overline{\mu}_i$
1	336	510	470
2	337	337	337
3	336	530	470
4	337	510	480
5	337	530	490

Table 2: The values of $\overline{q}$.

i	$\overline{q}_i^l$	$\overline{q}_i^u$	$\overline{q}_i$
1	19	90	54.5
2	19	19	19
3	19	29	39
4	19	183	101
5	19	103	61

In the elliptical case, the values of $\overline{\mu}$ and $\overline{q}$ are presented in Tables 1 and 2.

In order to solve the model, both cases of the box and robust elliptical counterparts are optimized separately. Moreover, for the binary variables of integer, the bifurcation and delimitation method is applied. The model shows that the main station is constructed at the potential location $j = 2$ and the difference between the models is in the number of ambulances allocated. In the robust counterpart formulation, the model also operates based on quadratic cone optimization. In the box and elliptic cases, the model of the main problem goes out of the semi-infinite state, and the robust state must ensure the existence of an optimal global solution in the quadratic linear and conical states [17]. In addition, because the research problem involves a constraint that all applications must be covered, all requests are covered by station $j = 2$. The total number of vehicles required at the station $j = 2$ in all three definite, box, and elliptical cases are 434, 478, and 367, respectively. Moreover, the percentage of demand that is covered in location i by station j in all three definite, box, and elliptic modes is $X_{1,2} = X_{2,2} = X_{3,2} = X_{4,2} = X_{5,2} = 1$. In the following subsection, the effect of each parameter on the objective functions is examined separately.

3.2. Comparing the Objective Functions. Pareto solutions can be obtained by using the weighting method [20] and by assigning a weight commensurate with the objective functions. By combining the two-objective functions (1) and (11), it can be stated that

$$\min \sum_{j\in J} f_j Y_j + \sum_{j\in j} P_j N_j + \sum_{j\in J} ct_j + W\sum_{j\in J}\widetilde{t}_j + \sum_{l} H_l S_l. \tag{49}$$

The performance evaluation of the proposed model is done by assigning different weights as well as the different number of candidate stations. The number of candidate stations for the medical emergency system is 30, 50, and 70, respectively. As mentioned earlier, all data used in this study are taken previously. Moreover, in cases where explicit data were not available, ten experiments were performed in

Table 3: The results of Model *P*.

	I	J	L	Q
0	11200	10050	10050	10050
5%	12500	10121	10190	10326
10%	13700	10163	10240	10562
15%	15200	10187	10412	11027
20%	17900	10239	10559	11276
25%	19600	10282	10612	11829
30%	19900	10355	10758	12414
35%	20050	10361	10908	12875
40%	20170	10396	10974	13141
45%	20200	10480	11023	13730
50%	20250	10546	11101	14107

Table 4: The results of Model *O*.

	I	J	L	Q
0	310200	475900	310300	295000
5%	327155	496635	310608	305520
10%	372825	840904	314181	312284
15%	395995	1271308	316860	316612
20%	475070	1309447	320803	321355
25%	507881	1348731	322527	330476
30%	558436	1389192	328203	333022
35%	585908	1430868	329706	335099
40%	667941	1473794	335171	335858
45%	681792	1518008	336705	345647
50%	782767	1563548	339967	357569

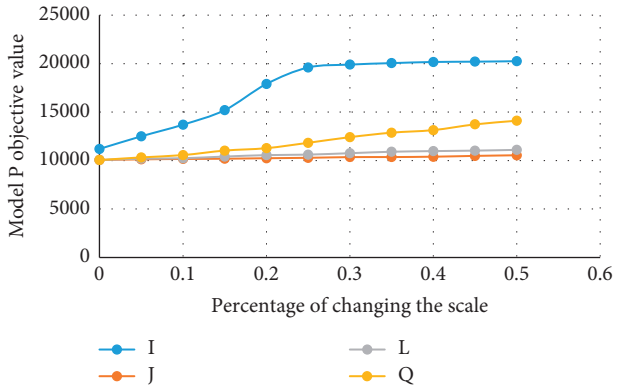


Figure 2: The effect of increasing the scale of the problem on the objective function of Model *P*.

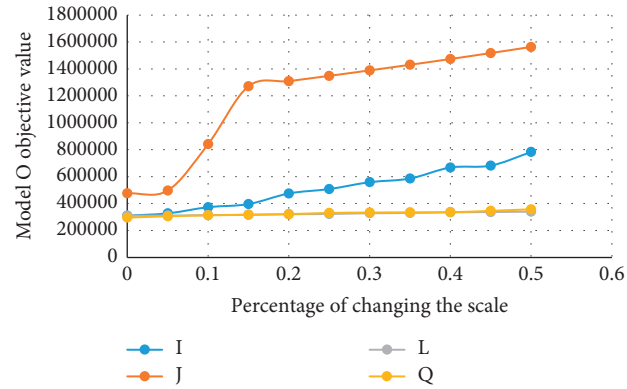


Figure 3: The effect of increasing the scale of the problem on the objective function of Model *O*.

which the parameters were obtained by multiplying the numbers obtained from the previous data in an interval $(1 + \varepsilon)$ where ε was obtained from a uniform distribution $[-0.1, 0.1]$. For more information, see [3, 5, 21–23]. The average speed of a car is 30 km/h. The average for each emergency job is one hour. The weight of uncovered demand is estimated at 50. The daily cost of the building at station j, i.e., f_j, is 50 units, and the cost of purchasing and maintaining each ambulance unit at station j, i.e., p_j, is two units. T also takes 15 minutes. The cost of each transport unit is estimated at 50. The maximum number of applications in the demand places is 10 units. The standard distance for applying the intelligent traffic management system is 200 units, and the cost of each unit of the traffic management system is 1 unit. According to the same demand that is considered for all three cases $j = 30$, 50, and 70, it is observed that the total number of ambulances required in different stations and in all three cases is 500 units and also increasing the number of stations from 30 up to 70 leads to an increase in costs, which can be fully seen in Table 3 and Figure 2.

Moreover, if an increasing trend can be found in the demand parameter, both the number of required ambulances and the value of objective function have increased significantly. By increasing the number of traffic light locations to apply the intelligent traffic management system, the objective function is first increased and then decreased. This may be due to the fact that as the number of intersections increases and, as a result, the traffic lights increase, the time for the ambulance to arrive from the station to the scene of the accident or from the place of demand to the hospital also increases and naturally leads to reduced response quality. Due to the lack of sufficient capacity in urban and intercity routes, costs increase. It is possible to increase the number of lights until there is a suitable platform for the application of an intelligent traffic management system, and therefore, in the absence of this, the system is not able to respond to all requests in a timely manner.

Finally, the overall cost of the system is reduced. Moreover, as the number of simultaneous demands increases, in theory, system costs and the number of ambulances required increase dramatically to meet overall demand. It should be noted that the impact of other parameters on the objective function, such as the daily cost of the emergency station, the daily cost of emergency vehicles, and the daily cost of an intelligent traffic management system, as well as vehicle speed, will increase the objective function, which, due to this increase, leads to no significant changes made to the objective function; this is not used in the following analysis. The set of changes mentioned is illustrated in Table 4 and Figure 3.

Moreover, the graph of the effect of the parameters on the number of ambulances required in both objective functions is as follows, in which the number of ambulances required changes with the change of the desired parameters,

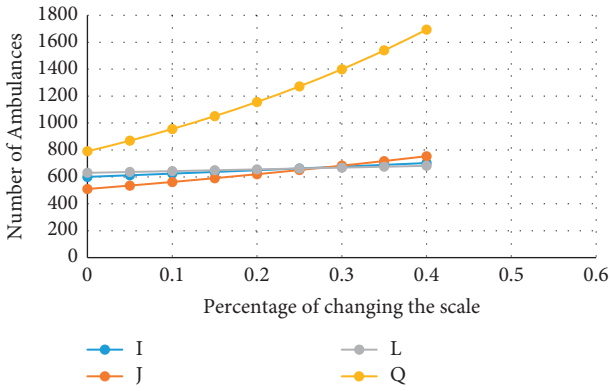


Figure 4: The effect of increasing the scale of the problem on the number of ambulances.

but, in the case of increasing the number of demand places, this change is more intense. And, in a way, it can be said that more ambulances are needed, which is possible by increasing the cost or increasing the objective function. Figure 4 shows the changes in the number of ambulances and their effect on both objective functions.

4. Conclusion and Future Works

In designing a two-objective medical emergency system, considering the intelligent traffic management system, the parameters that include the demand are uncertain. To guide this important matter, an approach called robust optimization has been used. Despite the overly conservative solution in the box uncertainty mode, both this mode and the elliptical uncertainty have been used. The robust counterpart and its function were introduced, and the robust counterpart approach was converted to linear programming and quadratic programming. This means that the robust counterpart provides an optimal global solution.

Nowadays, home care programs have increased in response to the needs of patients at the community level; one of the reasons for this rapid growth is the proof of the efficiency of home care for patients in the face of different needs of patients, which has been proven by increasing the efficiency and effectiveness of this method. A study of 600 patients showed that 81% of patients referred for hospitalization could be successfully treated at home. In addition, hospital stays were reduced from approximately 12 days to 7 days, and only 12% of patients in home care needed to be hospitalized again. The widespread use of research tools in the field of healthcare in today's world indicates the effectiveness of these tools. There are several benefits to serving patients at home today. According to the given definition of patient visit activities, it is possible to provide medical services and patient care and health care at home.

By optimizing the provision of home health services, patients can benefit from such manufacturing services. These services can include a wide range of activities, such as a doctor's patient visit, delivery of medicine and medical equipment to the patient, and receiving laboratory samples and medicines. Moreover, equipment has not been used to provide maintenance services for medical equipment and devices at home. In countries where we see more resistance to infectious diseases due to the high consumption of antibiotics, the patient can develop more resistance to infectious diseases by going to the hospital. On the other hand, the cost of providing medical services at home is much lower than the high costs of hospitalization and the country's healthcare system.

It shows that the best way to provide home care is not to prevent nosocomial infections. Other benefits include a significant reduction in healthcare costs, a reduction in the number of intermediaries and timely and efficient service, and the avoidance of wasting time. Reducing the traffic load of the city and more patient comfort in such activities should not be underestimated.

In order to indicate the future outlooks, it is suggested to apply a multifactor robust optimization approach to deal with the uncertainty. Moreover, it can also include studies that consider risk management in medical emergency systems. Furthermore, some novel solution methods like gray wolf optimizer and red dear algorithm can be used. On the other hand, in the case of the intelligent traffic management system, some new technologies like the Internet of Things (IoT) can be considered in the proposed mathematical model.

Data Availability

The data used are included in the article. Codes are also available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] V. Bélanger, A. Ruiz, and P. Soriano, "Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles," *European Journal of Operational Research*, vol. 272, no. 1, pp. 1–23, 2019.

[2] W. F. Dick, "Anglo-American vs. Franco-German emergency medical services system," *Prehospital and Disaster Medicine*, vol. 18, no. 1, pp. 29–37, 2003.

[3] S. J. Sajadi and A. Ahmadi, "An integrated optimization model and metaheuristics for assortment planning, shelf space allocation, and inventory management of perishable products: a real application," *PLoS One*, vol. 17, no. 3, Article ID e0264186, 2022.

[4] F. Goodarzian, A. A. Taleizadeh, P. Ghasemi, and A. Abraham, "An integrated sustainable medical supply chain network during COVID-19," *Engineering Applications of Artificial Intelligence*, vol. 100, Article ID 104188, 2021.

[5] P. Ghasemi, H. A. Khalili, A. P. Chobar, S. Safavi, and F. M. Hejri, "A New Multiechelon Mathematical Modeling for Pre-and Post disaster Blood Supply Chain: Robust Optimization Approach," *Discrete Dynamics in Nature and Society*, vol. 2022, Article ID 2976929, 2022.

[6] A. Goli and B. Malmir, "A covering tour approach for disaster relief locating and routing with fuzzy demand," *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 1, pp. 140–152, 2020.

[7] S. M. Pahlevan, S. M. S. Hosseini, and A. Goli, "Sustainable supply chain network design using products' life cycle in the aluminum industry," *Environmental Science and Pollution Research*, pp. 1–25, 2021.
[8] Z. H. Zhang and K. Li, "A novel probabilistic formulation for locating and sizing emergency medical service stations," *Annals of Operations Research*, vol. 229, Article ID 813835, 2015.
[9] S. Nickel, M. Reuter-Oppermann, and F. Saldanha-da-Gama, "Ambulance location under stochastic demand: a sampling approach," *Operations Research for Health Care*, vol. 8, pp. 24–32, 2016.
[10] R. Boujemaa, A. Jebali, S. Hammami, A. Ruiz, and H. Bouchriha, "A stochastic approach for designing two-tiered emergency medical service systems," *Flexible Services and Manufacturing Journal*, vol. 30, no. 1, pp. 123–152, 2017.
[11] C. Revelle, "Review, extension and prediction in emergency service siting models," *European Journal of Operational Research*, vol. 40, no. 1, pp. 58–69, 1989.
[12] H. Jia, F. Ordóñez, and M. Dessouky, "A modeling framework for facility location of medical services for large-scale emergencies," *IIE Transactions*, vol. 39, no. 1, pp. 41–55, 2007.
[13] C. Araz, H. Selim, and I. Ozkarahan, "A fuzzy multi-objective covering-based vehicle location model for emergency services," *Computers & Operations Research*, vol. 34, no. 3, Article ID 705726, 2007.
[14] J. Goldberg, "Operations research models for the deployment of emergency services vehicles," *EMS Management Journal*, vol. 1, no. 1, pp. 20–39, 2004.
[15] M. Daskin, "Network and discrete location: models, algorithms and applications," *Journal of the Operational Research Society*, vol. 48, no. 7, pp. 763-764, 1997.
[16] Z.-J. M. Shen and M. S. Daskin, "Trade-offs between customer service and cost in integrated supply chain design," *Manufacturing & Service Operations Management*, vol. 7, no. 3, pp. 188–207, 2005.
[17] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, Princeton University Press, USA, 2009.
[18] F. Goodarzian, H. Hosseini-Nasab, J. Muñuzuri, and M.-B. Fakhrzad, "A multi-objective pharmaceutical supply chain network based on a robust fuzzy model: a comparison of meta-heuristics," *Applied Soft Computing*, vol. 92, Article ID 106331, 2020.
[19] M. Ndiaye and H. Alfares, "Modeling health care facility location for moving population groups," *Computers & Operations Research*, vol. 35, no. 7, pp. 2154–2161, 2008.
[20] J. Cohon, *Multiobjective Programming and Planning*, Dover Pubns, USA, 2004.
[21] A. Gaspar, D. Oliva, S. Hinojosa, I. Aranguren, and D. Zaldivar, "An optimized Kernel extreme learning machine for the classification of the autism spectrum disorder by using gaze tracking images," *Applied Soft Computing*, vol. 120, Article ID 108654, 2022.
[22] S. Ganguly, P. Bhowal, D. Oliva, and R. Sarkar, "BLeafNet: a Bonferroni mean operator based fusion of CNN models for plant identification using leaf image classification," *Ecological Informatics*, vol. 69, Article ID 101585, 2022.
[23] A. Pourghader Chobar, M. A. Adibi, and A. Kazemi, "A novel multi-objective model for hub location problem considering dynamic demand and environmental issues," *Journal of Industrial Engineering and Management Studies*, vol. 8, no. 1, pp. 1–31, 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 3082933, 19 pages
https://doi.org/10.1155/2022/3082933

Research Article

Application of Improved Manta Ray Foraging Optimization Algorithm in Coverage Optimization of Wireless Sensor Networks

Fang Zhu, Wenhao Wang, and Shan Li

School of Computer and Communication Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

Correspondence should be addressed to Wenhao Wang; 2072030@stu.neu.edu.cn

Received 17 February 2022; Revised 1 June 2022; Accepted 18 June 2022; Published 30 June 2022

Academic Editor: Seyed Jalaleddin Mousavirad

For the shortcomings of the manta ray foraging optimization (MRFO) algorithm, like slow convergence speed and difficult to escape from the local optimum, an improved manta ray foraging algorithm based on Latin hypercube sampling and group learning is proposed. Firstly, the Latin hypercube sampling (LHS) method is introduced to initialize the population. It divides the search space evenly so that the initial population covers the whole search space to maintain the diversity of the initial population. Secondly, in the exploration stage of cyclone foraging, the Levy flight strategy is introduced to avoid premature convergence. Before the somersault foraging stage, the adaptive t-distribution mutation operator is introduced to update the population to increase the diversity of the population and avoid falling into the local optimum. Finally, for the updated population, it is divided into leader group and follower group according to fitness. The follower group learns from the leader group, and the leader group learns from each other through differential evolution to further improve the population quality and search accuracy. 15 standard test functions are selected for comparative tests in low and high dimensions. The test results show that the improved algorithm can effectively improve the convergence speed and optimization accuracy of the original algorithm. Moreover, the improved algorithm is applied to wireless sensor network (WSN) coverage optimization. The experimental results show that the improved algorithm increases the network coverage by about 3% compared with the original algorithm, and makes the optimized node distribution more reasonable.

1. Introduction

With the advancement and development of intelligent information technology, the scale and complexity of data are also increasing. Traditional numerical optimization methods are difficult to solve complex optimization problems, resulting in higher and higher calculation costs. In recent years, swarm-based intelligent optimization algorithms have been favored by many researchers because of their simplicity and high efficiency [1]. Swarm intelligence algorithms can effectively solve many complex optimization problems in the field of engineering, and are mainly used in network optimization [2], feature selection [3], image processing [4], automatic control [5], and other fields. In recent years, swarm intelligence optimization algorithms have been proposed, including butterfly optimization algorithm (BOA) [6], whale optimization algorithm (WOA) [7], sine cosine algorithm (SCA) [8], sparrow search algorithm (SSA) [9], marine predator algorithm (MPA) [10], African vultures optimization algorithm (AVOA) [11], manta ray foraging optimization (MRFO) algorithm [12], and so on.

The MRFO algorithm is a new swarm intelligence optimization algorithm proposed by Weiguo Zhao et al. in 2020. The inspiration of this algorithm is based on intelligent behaviors of manta rays. The foraging (optimization) process is divided into three stages, namely chain foraging, cyclone foraging, and somersault foraging. Compared with some classical intelligent algorithms and most of the above algorithms, it has higher convergence accuracy and faster optimization speed. Although MRFO has the above advantages, it still has the problems of easy premature convergence and falling into local optimum. In order to solve these problems, many researchers have improved the basic MRFO algorithm. Davut Izci et al. [13] introduced the opposition-based learning strategy into the population initialization, which improves the quality of the population to a certain extent, but

convergence accuracy needs to be improved. Biqi Sheng et al. [14] proposed a balanced manta ray foraging optimization (BMRFO) algorithm. BMRFO introduces the Levy flight strategy in the cyclone foraging stage, and improves the flip factor. Although the algorithm's ability to jump out of the local optimum is improved, the convergence speed is not significantly improved. Oguz [15] introduces the chaotic map into the foraging behavior of MRFO, which improves the optimization performance of the algorithm, but the improvement ability was limited.

In order to better solve the problems and improve the optimization accuracy and convergence speed of the MRFO algorithm, this paper combines the Latin hypercube sampling (LHS) method with the group learning strategy, and introduces the Levy flight and adaptive t-distribution disturbance strategy. Therefore, an improved MRFO algorithm based on LHS and group learning (LGMRFO) is proposed. To verify the performance of the LGMRFO algorithm, 15 general test functions and 9 CEC2017 test suite functions are selected for low-dimensional and high-dimensional comparison tests.

Adaptive adjustment and deployment of sensor nodes in WSN can make them more evenly distributed in the detection area and have a higher coverage, so as to rationally allocate network space resources and better complete the tasks of environmental awareness and information acquisition. This is of great significance to improve network viability, improve network reliability, and save network construction costs. Generally, area coverage is the main criterion for evaluation. Optimized coordinate deployment of sensor nodes is carried out through optimization algorithm, and as few sensor nodes as possible are used to ensure the area coverage requirement and reduce the redundancy of sensor nodes. Therefore, in order to improve the poor coverage effect caused by unreasonable deployment of WSN nodes, the LGMRFO is applied to the coverage optimization problem of WSN. The experimental results further verify the effectiveness of the algorithm.

The rest of this paper is organized as follows. The MRFO algorithm is described in details in section "MRFO". Section "Related Works" detailly introduces some intelligence optimization algorithms. Section "LGMRFO" describes the improved strategies for MRFO in this work. The performance of LGMRFO is evaluated by optimizing 24 test functions in section "Numerical Simulation Analysis". Section "Coverage optimization of WSN Based on LGMRFO" presents the simulations and performance evaluation of LGMRFO for WSN coverage. At last, Section "Conclusion" summarizes this paper.

2. Related Works

Based on the source inspiration, the intelligence optimization algorithms can be divided into four classes of [16]: (a) physics-based, (b) math-based, (c) human-based, and (d) swarm-based. Physics-based methods tend to perceive the landscape as a physical phenomenon and move the search agents using formulae borrowed from physical rules or theories. The Archimedes optimization algorithm [17] is devised with inspirations from an interesting law of physics Archimedes' principle. An equilibrium optimizer (EO) [18] is inspired by control volume mass balance models used to estimate both dynamic and equilibrium states. Atomic orbital search (AOS) [19] is proposed based on some principles of quantum mechanics and the quantum-based atomic model. Transient search optimization (TSO) [20] is inspired by the transient behavior of switched electrical circuits that include storage elements.

Math-based algorithms are solely based on mathematical equations. They are not inspired by a specific natural phenomenon. Runge Kutta optimizer (RUN) [21] is designed according to the mathematical foundations of the Runge Kutta method. Gradient-based optimizer (GBO) [22] is inspired by the gradient-based Newton's method. The golden sine algorithm (Gold-SA) [23] is inspired by sine that is a trigonometric function. The arithmetic optimization algorithm (AOA) [24] utilizes the distribution behavior of the main arithmetic operators in mathematics. Weighted mean of vectors (INFO) [25] is an efficient optimization algorithm based on weighted mean of vectors.

Inspired by the social behaviors of human beings, a lot of optimization algorithms have been proposed. Political optimizer (PO) [26] is inspired by the multiphased process of politics. The group teaching optimization algorithm (GTOA) [27] simulated the impact of teachers on learners' output in the classroom. Queuing search (QS) [28] is inspired from human activities in queuing. Student psychology based optimization (SPBO) [29] is inspired by the psychology of the students who are trying to give more effort to improve their performance in the examination up to the level for becoming the best student in the class.

Swarm-based approaches imitate the social behavior and communications within a group of species of animals, plants, or other living things. These approaches have gained increasing popularity in terms of both application and new algorithm development. Some of the recently proposed algorithms that can be categorized under this approach are slime mould algorithm (SMA) [30], hunger games search (HGS) [31], Harris hawks optimization (HHO) [32], moth search algorithm (MSA) [33], monarch butterfly optimization (MBO) [34], golden eagle optimizer (GEO) [35], and tuna swarm optimization (TSO) [36].

Compared with these three types, swarm-based algorithms have superiority over other three types of algorithms. Manta ray foraging optimization (MRFO), with few adjustable parameters, is easy to implement, which in turn makes it very potential for applications in many engineering fields. So, this paper improves the manta ray foraging optimization (MRFO) algorithm named MRFO based on Latin hypercube sampling and group learning (LGMRFO). MRFO falls into the fourth class of optimization algorithms, as it originates from swarm behavior of manta rays (a kind of sea animal).

3. MRFO

MRFO updates the individual position by three foraging behaviors, including chain foraging, cyclone foraging, and

somersault foraging. The mathematical models are described below.

3.1. Chain Foraging. Manta rays' line up head-to-tail and form a foraging chain. In each iteration, each individual is updated by the best solution found so far and the solution in front of it. This mathematical model of chain foraging is represented as follows:

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 1, \\ x_i^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 2, \ldots, N, \end{cases} \tag{1}$$

$$\alpha = 2 \cdot r \cdot \sqrt{|\log(r)|}, \tag{2}$$

where, $x_i^d(t)$ is the position of ith individual at t-th iteration, r is a random vector within the range of [0, 1], α is a weight coefficient, $x_{\text{best}}^d(t)$ is the plankton with high concentration (the best solution found so far), and N denotes the population size.

3.2. Cyclone Foraging. When manta rays find plankton in deep water, they form a long foraging chain and swim towards the food by a spiral. In the cyclone foraging behavior of manta rays, in addition to spirally move towards the food, each manta ray swims towards the one in front of it. The mathematical model of the exploitation stage of cyclone foraging behavior can be calculated by the following formula:

$$x_i^d(t+1) = \begin{cases} x_{\text{best}}^d + r \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 1, \\ x_{\text{best}}^d + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 2, \ldots, N, \end{cases} \tag{3}$$

$$\beta = 2e^{r_1(T-t+1/T)} \cdot \sin(2\pi r_1), \tag{4}$$

where β is a weight factor, T is the maximum number of iterations, and r_1 is a rand number in [0, 1].

In equation (3), MRFO focuses on local exploitation. In addition, by taking the random position in the search space as the reference position, this behavior can also be used to improve the exploration mechanism of the algorithm. The mathematical model is as follows:

$$x_{\text{rand}}^d = Lb^d + r \cdot \left(Ub^d - Lb^d\right),$$
$$x_i^d(t+1) = \begin{cases} x_{\text{rand}}^d + r \cdot \left(x_{\text{rand}}^d - x_i^d(t)\right) + \beta \cdot \left(x_{\text{rand}}^d - x_i^d(t)\right), & i = 1, \\ x_{\text{rand}}^d + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{rand}}^d - x_i^d(t)\right), & i = 2, \ldots, N, \end{cases} \tag{5}$$

where x_{rand}^d is a random position produced in the search space, Lb^d and Ub^d are the lower and upper limits of the dth dimension, respectively.

3.3. Somersault Foraging. In this foraging behavior, the position of food is regarded as a pivot. Each individual tends to swim to and from around the pivot and somersault to a new position. The mathematical model can be created as follows:

$$x_i^d(t+1) = x_i^d(t) + S \cdot \left(r_2 \cdot x_{\text{best}}^d - r_3 \cdot x_i^d(t)\right), \quad i = 1, \ldots, N, \tag{6}$$

where S is the somersault factor that decides the somersault range of manta rays and $S = 2$, r_2 and r_3 are two random number in [0, 1].

MRFO balances the ability of global exploration and local exploitation by controlling the change in t/T, where, t is the current number of iterations and T is the maximum number of iterations. When $t/T <$ rand, selecting the current optimal position as the reference position for global exploration behavior. When $t/T \geq$ rand, taking the optimal individual as the reference point, it focuses on the local exploitation ability of the algorithm.

4. LGMRFO

In order to improve the performance of MRFO, this paper improves it in three aspects: Firstly, the LHS method is used to initialize the population to enhance the diversity of the population; Secondly, in the exploration stage of cyclone foraging, Levy flight strategy is introduced to accelerate the convergence speed. Before the somersault foraging, an adaptive t-distribution mutation operator is added to update the population position to avoid falling into local optimization; Finally, the group learning strategy is set to improve the optimization accuracy of the algorithm.

4.1. LHS Method Population Initialization Strategy. In the basic MRFO, the initial population is generated in a random way. The initial population generated by this method is often unevenly distributed or even overlaps individuals, which reduces the optimization performance of the algorithm to a certain extent. The LHS method is a multidimensional stratified sampling technology proposed by McKay et al. [37], which has the following advantages compared with simple random sampling method.

(1) The sampling points generated by LHS can achieve full space coverage and can be evenly distributed in the search space;

(2) LHS has better robustness and stability.

Therefore, in order to enhance the diversity of the initial population and improve the performance, we adopt the LHS method to initialize the population.

Assuming that N initial individuals are generated in the d-dimensional space, the specific steps to initialize the population with the LHS method are as follows:

Step 1. Firstly, the population size N and dimension d are determined.

Step 2. Determine the interval for individual x as [lb, ub], where lb and ub are the lower and upper bounds of the variable x, respectively.

Step 3. Divide the interval of variable x into N equal small intervals.

Step 4. Randomly select a point in each subinterval of each dimension.

Step 5. Combine the extracted points of each dimension to form initial population.

Figure 1 and 2 are sample point maps generated by the LHS method and simple random sampling method, respectively, where the sampling size is 20 and dimension is 2. It can be seen that the sample points generated by the LHS method can be more evenly distributed in the search space. Therefore, using the LHS method to initialize the population of the MRFO algorithm, it can make the population position evenly distributed in the search space, and enhance the population diversity to improve the convergence performance of the algorithm.

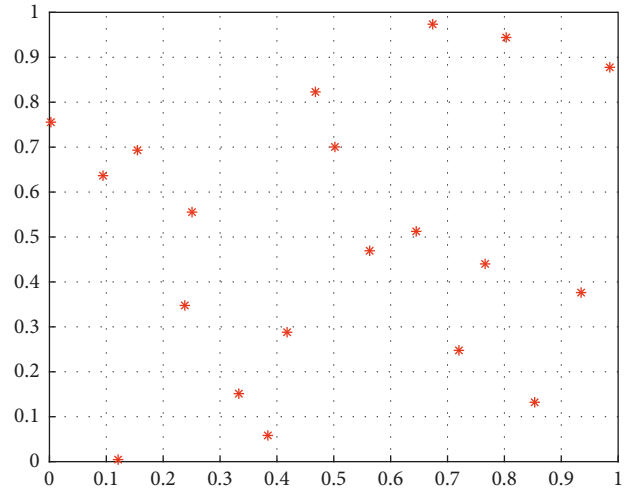


Figure 1: Samples distribution map based on the LHS method.

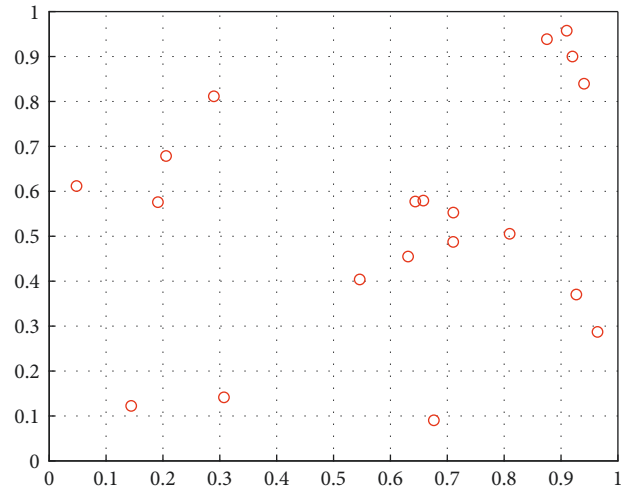


Figure 2: Samples distribution map based on the random method.

4.2. Mutation Strategy

4.2.1. Levy Flight. In some cases, due to the random individual selection in each iteration, premature convergence may occur, thereby increasing the running time, so different mechanisms can be used to improve the MRFO algorithm. This paper uses the Levy flight mechanism [38] for local disturbance, the mechanism is based on random walk behavior, and the mathematical model is as follows:

$$\text{Levy}(\lambda) = \frac{u}{|v|^{-\lambda}}, \tag{7}$$

where u and v come from the normal distribution, i.e.,

$$u \sim N\left(0, \sigma_u^2\right) v \sim N\left(0, \sigma_v^2\right). \tag{8}$$

The values of σ_u and σ_v are as follows:

$$\sigma_u = \left\{\frac{\Gamma(1+\lambda)\sin(\pi\lambda/2)}{\Gamma(1+\lambda/2)\lambda 2^{\lambda-1/2}}\right\}^{1/\lambda}, \quad \sigma_v = 1, \tag{9}$$

where Γ is the standard gamma function.

The position update formula of the cyclone foraging exploration stage with the addition of Levy flight strategy is as follows:

$$x_i^d(t+1) = \begin{cases} x_{\text{rand}}^d + \text{Levy}(\lambda) \otimes \left[r \cdot \left(x_{\text{rand}}^d - x_i^d(t)\right) + \beta \cdot \left(x_{\text{rand}}^d - x_i^d(t)\right)\right], & i = 1, \\ x_{\text{rand}}^d + \text{Levy}(\lambda) \otimes \left[r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{rand}}^d - x_i^d(t)\right)\right], & i = 2, \ldots, N, \end{cases} \tag{10}$$

where $\otimes$ denotes point-to-point multiplication.

4.2.2. Adaptive t-Distribution. T-distribution is also called student distribution [39], and its distribution state is closely related to degrees of freedom. In order to enhance the diversity of the population and avoid falling into local optimum, this paper introduces the adaptive t-distribution strategy to disturb manta ray population before the somersault foraging behavior. The calculation formula is as follows:

$$x_{\text{new}} = x_{\text{old}} + x_{\text{old}} \cdot t(\text{iter}), \tag{11}$$

where x_{old} is the original individual, x_{new} is the new individual after mutation, and $t(\text{iter})$ is the t-distribution with the current iteration number iter as the degree of freedom.

In the early stage of the iteration, the degree of freedom is small (the number of iterations is small), and t-distribution is similar to Cauchy distribution. At this time, the update step size is larger, which can expand the search field of the individual and improve the global exploration ability. In the middle and later iteration, the degree of freedom gradually increases, and the performance of t-distribution is similar to Gauss distribution. At this time, the update step size is smaller, which helps the algorithm to search around the current individual neighbourhood, and the algorithm has better local exploitation ability.

4.3. Group Learning Strategy. In the process of algorithm evolution, some individuals may reach the optimal position, and the fitness value of others may become more worse. In order to overcome this defect, inspired by the salp swarm algorithm (SSA) [40], individuals with poor location need to learn foraging skills from individuals with good location. Based on this idea, a group learning strategy is proposed. The population after somersault foraging is evenly divided into two groups according to the fitness value. The group with better fitness is called the leader group, and the group with poor fitness is called the follower group.

4.3.1. Leader Group Learning Strategy. The differential evolution (DE) algorithm [41] has a good effect in solving complex optimization problems. In this paper, the differential evolution strategy is used to generate a new leader group individual, and the greedy strategy is used to select the optimal individual. The specific mathematical model is as follows:

$$x_{\text{new}} = x'_{\text{best}} + F \cdot (x_m - x_n), \tag{12}$$

where x_{new} is a new individual produced by mutation; x'_{best} is the optimal individual x_{best} new individuals generated by randomly sorting dimensions; F is the scaling factor, and $F = 0.5$; x_m and x_n are two different leaders randomly selected from the leadership group, which are different from the current individual. The new individual generated by this strategy needs to be compared with the original individual, and the individual with better fitness should be selected as the current individual.

Compared with the mutation of whole individuals, this strategy has stronger selectivity, which can effectively enhance the local mining performance and improve the convergence accuracy of the algorithm.

4.3.2. Follower Group Learning Strategy. Each follower in the follower group learns from the average of the two leaders. The mathematical model is described as follows:

$$x_{\text{follower}}^{i\_\text{new}} = \frac{\left(x_{\text{leader}}^i + x_{\text{leader}}^{i+1}\right)}{2}, \tag{13}$$

where $x_{\text{follower}}^{i\_\text{new}}$ refers to the new individual generated after the ith individual of the following group learns from the leading group, x_{leader}^i represents the ith individual of the leadership group. The new follower individual needs to be compared with the original follower individual, and the individual with a better fitness value is selected as the current follower individual.

By learning from the leader group, the follower group can greatly improve the fitness, realize the conversion from follower to leader, and then improve the convergence speed of the algorithm.

4.4. LGMRFO Algorithm Implementation Steps. The specific implementation steps of LGMRFO algorithm are as follows:

Step 1. Set the relevant parameters: population size N, variable dimension D, maximum number of iterations T, and initialize the population position by the LHS method.

Step 2. The fitness value of each individual is calculated, and the initial optimal individual position and its optimal fitness value are obtained according to the fitness value.

Step 3. Enter the algorithm iteration process. When rand ≥ 0.5, chain foraging is performed and updates the individual position according to equation (1); Otherwise,

Input: Initialize the size of population N, the maximal number of iterations T, and the manta rays X.
Output: The best solution Xbest.
(1) Compute the fitness of each individual fi = f(Xi) and obtain the best solution found so far Xbest, where lb and ub are the lower and upper boundaries of problem space, respectively.
(2) Initialize the iteration counter $t = 0$
(3) While $t < T$
(4) For $i = 1$ to N
(5) if rand <0.5
(6) if t/T < rand
(7) Perform the exploratory behavior of cyclone foraging according to equation (10)
(8) else
(9) Perform the exploitative behavior of cyclone foraging according to equation (3)
(10) end
(11) else
(12) Perform the chain foraging according to equation (1)
(13) end
(14) end
(15) Greedy selection of the current individual
(16) For $i = 1$ to N
(17) Perform the adaptive t-distribution strategy according to equation (11)
(18) end
(19) Greedy selection of the current individual
(20) For $i = 1$ to N
(21) Perform the somersault foraging according to equation (6)
(22) end
(23) The population was divided into two groups according to the fitness value
(24) Perform the group learning strategy according to equations (12) and (13)
(25) Compute the fitness of each individual $f_i = f(X_i)$ and obtain the best solution found so far Xbest
(26) $t = t + 1$
(27) end
(28) Return the best solution found so far Xbest

Algorithm 1: LGMRFO Algorithm.

cyclone foraging is performed, when $t/T <$ rand, the individual enters the exploration stage, introduces Levy flight strategy, and updates the individual position according to equation (10), when $t/T <$ rand, the individual enters the development stage and updates the individual position according to equation (3).

Step 4. Before performing somersault foraging behavior, an adaptive t-distribution strategy is added, the individual position is updated according to equation (11), and the current individual is greedily selected.

Step 5. Perform somersault foraging behavior according to equation (6).

Step 6. The group learning strategy is implemented, that is, the updated population is divided into a leading group and a following group according to the fitness value, and new individuals are generated by learning from equations (12) and (13), respectively. If the fitness becomes better after learning, the current individual position will be updated, otherwise, it will not be updated.

Step 7. Update the optimal location and its optimal fitness value of each generation.

Step 8. Judge whether the algorithm meets the iteration conditions. If so, the algorithm terminates; Otherwise, go to Step 3.

The pseudocode of LGMRFO is shown in Algorithm 1.

4.5. Time Complexity of the LGMRFO. The overall time complexity of MRFO is given as

$$\begin{aligned} O(\text{MRFO}) &= O(T(O(\text{cyclone foraging} + \text{chain foraging}))) \\ &\quad + O(\text{somersault foraging}), \\ O(\text{MRFO}) &= O(T(nd + nd)) = O(Tnd), \end{aligned} \tag{14}$$

where, T is the maximum number of iterations, n is the number of individuals, and d is the number of variables.

LGMRFO proposed that in this paper only increases the computational complexity in adaptive t-distribution and group learning. Therefore, the overall time complexity of LGMRFO is given as

$$\begin{aligned} O(\text{LGMRFO}) &= O(\text{MRFO}) + O(T(\text{O}(\text{adaptive } t \\ &\quad - \text{distribution}) + O(\text{group learning}))), \\ O(\text{LGMRFO}) &= O(\text{MRFO}) + O(T(nd + nd)) = O(Tnd). \end{aligned} \tag{15}$$

This shows that the time complexity of LGMRFO is consistent with that of MRFO.

Table 1: Test function information.

No.	Function name	Function formula	D	Range	Optimum
F1	Bent cigar	$F1(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	50/500	[−100, 100]	0
F2	Sum of different power	$F2(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	50/500	[−100, 100]	0
F3	Zakharov	$F3(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$	50/500	[−100, 100]	0
F4	High conditioned	$F4(x) = \sum_{i=1}^{D} (10^6)^{i-1/D-1} x_i^2$	50/500	[−100, 100]	0
F5	Griewank's	$F5(x) = \sum_{i=1}^{d} x_i^2/4000 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	50/500	[−100, 100]	0
F6	Rastrigin	$F6(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	50/500	[−100, 100]	0
F7	Expanded Schaffer's	$F7(x) = \sum_{i=1}^{D-1} y(x_i, x_{i+1}) + y(x_D, x_1)$			
		$y(u, v) = 0.5 + (\sin^2(\sqrt{u^2 + v^2}) - 0.5)/(1 + 0.001(u^2 + v^2))^2$	50/500	[−100, 100]	0
F8	Noncontinuous rotated Rastrigin's	$F8(x) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$ $y_i = \begin{cases} x_i & \lvert x_i \rvert < 0.5 \\ \text{round}(2x_i)/2 & \lvert x_i \rvert \geq 0.5 \end{cases}$	50/500	[−100, 100]	0
F9	Rosenbrock's	$F9(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	50/500	[−100, 100]	0
F10	*Discus*	$F10(x) = 10^6 x_1^2 + \sum_{i=1}^{D} x_i^2$	50/500	[−100, 100]	0
F11	Ackley	$F11(x) = -20\exp(-0.21/D\sum_{i=1}^{D} x_i^2) - \exp(1/D\sum_{i=1}^{D} \cos(2\pi x_i)) + 20 + e$	50/500	[−100, 100]	0
F12	Schaffer's F7	$F12(x) = [1/D - 1\sum_{i=1}^{D-1} (\sqrt{s_i} \times (\sin(50 s_i^{0.2}) + 1))]^2$ $s_i = \sqrt{x_i^2 + x_{i+1}^2}$	50/500	[−100, 100]	0
F13	Foxholes	$F13(x) = (1/500 + \sum_{j=1}^{25} 1/j + \sum_{i=1}^{2} (x_i - a_{ij})^6)^{-1}$	2	[−65, 65]	1
F14	Schkel	$F14(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.5363
F15	Six-hump camel	$F15(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316

5. Numerical Simulation Analysis

24 test functions are chosen and experimentally tested with five algorithms for finding the minimal value of the function: BOA, WOA, SCA, SSA (sparrow search algorithm), and MRFO, in order to evaluate the effectiveness of the proposed LGMRFO algorithm. Refer to the corresponding original literature for specific parameter settings.

5.1. Test Functions

(1) General functions: Single-peaked functions have only one global best point and no local extreme points, while the test functions F1 to F4 are multidimensional single-peaked functions, F5 to F12 are high-dimensional multipeaked functions, and F13 to F15 are three fixed-dimensional multipeaked functions. The multipeak function has numerous local extremum points, which are utilized to observe the performance of the function jumping out of local extremum points in different dimensions from two high-dimensional views. Table 1 shows the precise function details.

(2) CEC2017 test suite functions: In order to further test the performance of LGMRFO, this paper selects some CEC2017 test suite functions [42] for testing, which are CF2, CF4, CF7, CF8, CF10, CF15, CF17, CF20, and CF24, respectively, with $D = 30$ and Range $\in [-100, 100]$.

5.2. Results Evaluation of General Functions. Simulation and comparison experiments of six algorithms were conducted in the Matlab R2018a environment. To avoid excessive chance errors, each benchmark function was chosen to run 30 times independently in the experiments, and the optimal value, the worst value, the average value, and standard deviation were used as evaluation indexes, and the population size was set to 30 and the maximum number of iterations was 500. Black highlights the greatest outcomes. F13 to F15 (fixed-dimensional multipeak function), $D = 50$ (low-dimensional), and $D = 500$ (high-dimensional) functions F1 to F12, respectively, are used to test and assess the algorithm.

The iterative convergence curves of six algorithms at 500 dimensions under four single peak test functions, eight multipeak function test functions, and three fixed-dimension test functions are plotted in this research due to the article length constraint, as shown in Figure 3.

5.2.1. Multipeak Function Test with Fixed Dimensions. Table 2 shows the test results for the three fixed-dimension multipeak functions from F13 to F15. Table 2 and Figures 3(e) and 3(f) show that LGMRFO has faster convergence and better optimization-seeking accuracy than other algorithms, and its standard deviation is the smallest, indicating that it is more stable. The standard deviation reflects the algorithm's stability in solving, so LGMRFO is more stable.

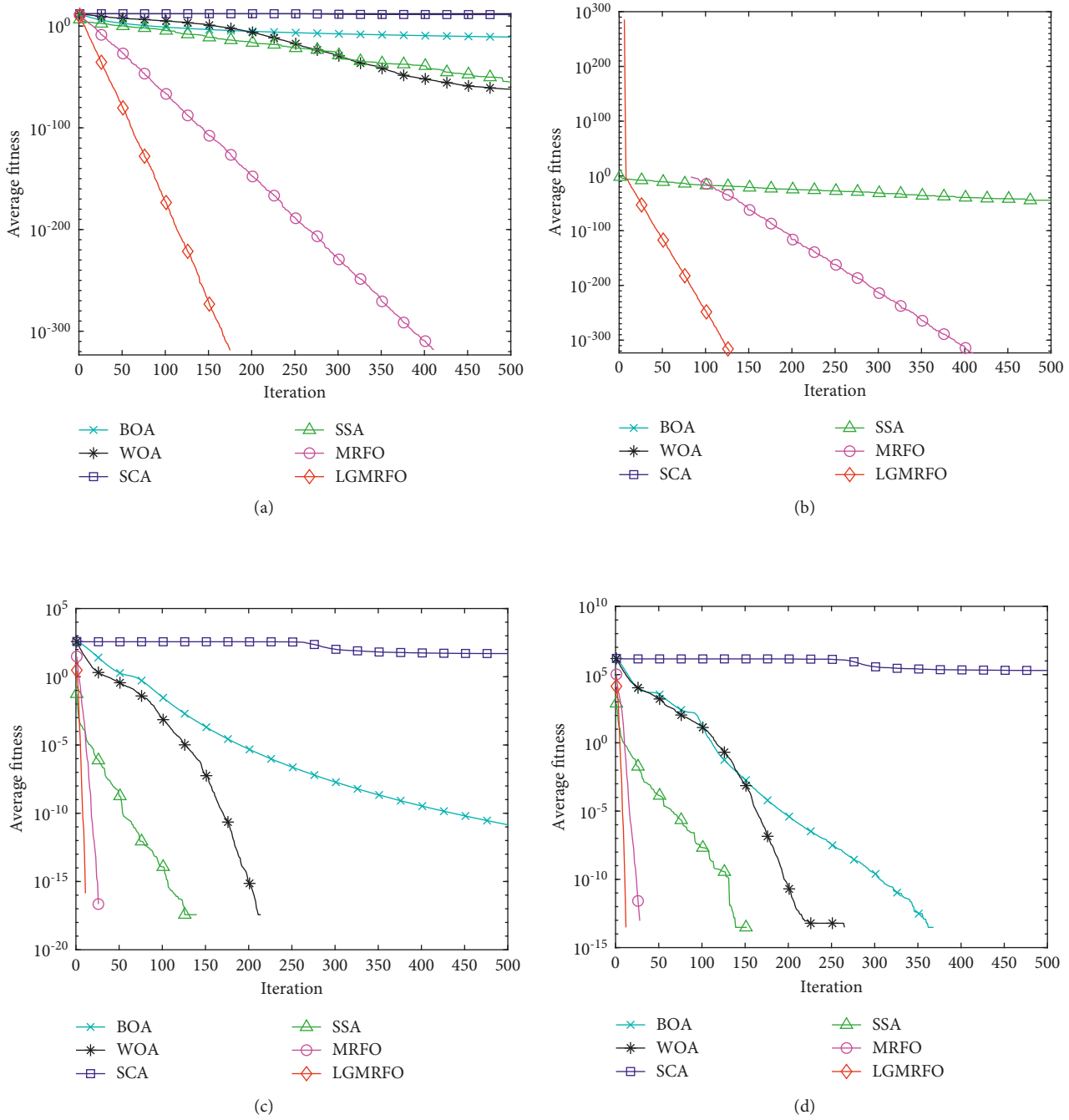


Figure 3: Continued.

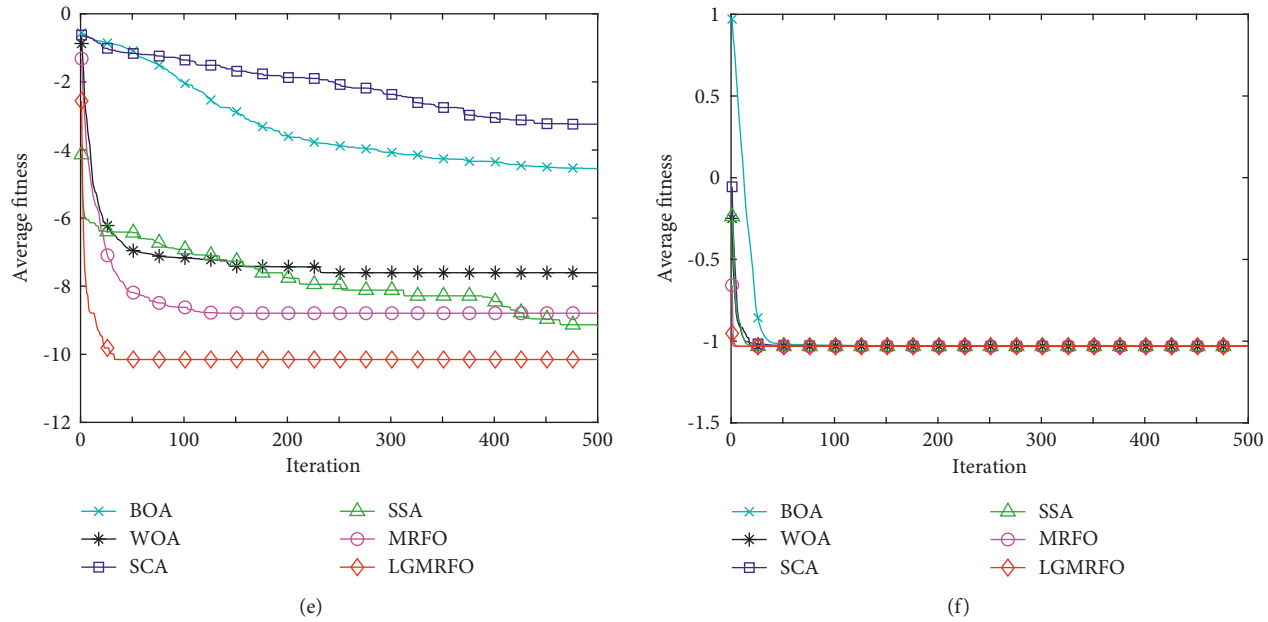


Figure 3: Average convergence curves of 500 dimensional partial functions and fixed-dimensional functions F14 and F15. (a) F1 (b) F2. (c) F5 (d) F6. (e) F14 (f) F15.

Table 2: Comparison of test results for fixed-dimensional function F13–F15.

Function	Algorithm	Best	Worst	Average	Std. Deviation
F13	BOA	0.9980	3.0050	1.2986	0.6261
	WOA	0.9980	10.7632	3.0888	3.5685
	SCA	0.9980	2.9821	1.7922	0.9882
	SSA	0.9980	12.6705	7.3556	5.7991
	MRFO	0.9980	0.9980	0.9980	$1.01E-16$
	LGMRFO	0.9980	0.9980	0.9980	$9.22E-17$
F14	BOA	−5.4578	−4.0632	−4.5494	0.2656
	WOA	−10.1531	−2.6283	−7.6042	2.8176
	SCA	−7.8812	−0.4973	−3.2410	2.0496
	SSA	−10.1532	−5.0552	−9.2006	1.9674
	MRFO	−10.1532	−5.0552	−8.7937	2.2930
	LGMRFO	−10.1532	−10.1532	−10.1532	$6.33E-15$
F15	BOA	−1.0316	−1.0287	−1.0307	$8.58E-04$
	WOA	−1.0316	−1.0316	−1.0316	$2.01E-09$
	SCA	−1.0316	−1.0316	−1.0316	$3.96E-05$
	SSA	−1.0316	−1.0316	−1.0316	$6.39E-16$
	MRFO	−1.0316	−1.0316	−1.0316	$6.52E-16$
	LGMRFO	−1.0316	−1.0316	−1.0316	$5.61E-16$

5.2.2. Evaluation of Low-Dimensional Functions. Table 3 shows a comparison of the algorithm's function test results in 50 dimensions. Both LGMRFO and basic MRFO can meet the theoretical optimal value in the single-peak low-dimensional function test, as shown in Table 3, and standard deviation is 0. This indicates that LGMRFO's optimization-seeking ability is more stable than other algorithms, and LGMRFO's convergence speed is significantly faster than other intelligent algorithms, including MRFO, indicating that the improvement strategy has significantly improved MRFO's convergence performance. Table 2 shows that LGMRFO can also get greater accuracy solutions in the multipeak low-dimensional function test, especially for functions F5, F6, F7, F8, and F10. Although the average solutions of functions F9, F11, and F12 do not approach the theoretical ideal value, LGMRFO's overall convergence performance ranks 2nd, 1st, and 2nd, respectively, when compared to other algorithms. The other algorithms have a better chance of escaping the local optimum. Except for functions F9 and F12, LGMRFO has the smallest standard deviation among the other functions, hence its robustness is higher in terms of stability.

Table 3: Comparison of test results under different dimensions for function F1–F12.

Function	Algorithm	$D = 50$				$D = 500$			
		Best	Worst	Average	Std	Best	Worst	Average	Std
F1	BOA	$1.38E-11$	$1.68E-11$	$1.55E-11$	$8.44E-13$	$1.45E-11$	$1.76E-11$	$1.60E-11$	$9.61E-13$
	WOA	$1.08E-79$	$1.32E-63$	$7.38E-65$	$2.86E-64$	$9.70E-81$	$2.86E-61$	$1.11E-62$	$5.23E-62$
	SCA	1.01 E+07	3.24 E+09	8.14 E+08	$8.47E+08$	$6.92E+10$	$3.38E+11$	$2.03E+11$	$7.12E+10$
	SSA	0	$1.20E-60$	$4.01E-62$	$2.19E-61$	$1.74E-248$	$2.91E-54$	$9.69E-56$	$5.31E-55$
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F2	BOA	$6.27E+74$	$9.03E+89$	$8.05E+88$	$2.18E+89$	Inf	Inf	Inf	NaN
	WOA	$4.34E-112$	$2.54E-82$	$9.08E-84$	$4.65E-83$	Inf	Inf	Inf	NaN
	SCA	$2.41E+24$	$4.86E+48$	$1.65E+47$	$8.87E+47$	Inf	Inf	Inf	NaN
	SSA	0	$2.99E-37$	$9.96E-39$	$5.46E-38$	0	$7.82E-44$	$2.61E-45$	$1.43E-44$
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F3	BOA	$1.11E-11$	$1.46E-11$	$1.28E-11$	$8.41E-13$	$1.10E-11$	$1.49E-11$	$1.34E-11$	$8.32E-13$
	WOA	$6.01E+04$	$1.47E+05$	$9.47E+04$	$1.83E+04$	$1.54E+06$	$1.86E+06$	$1.64E+06$	$6.22E+04$
	SCA	$2.643E+03$	$2.81E+04$	$1.28E+04$	$6.33E+03$	$2.98E+05$	$8.34E+05$	$5.59E+05$	$1.31E+05$
	SSA	$5.52E-169$	$1.24E-74$	$6.94E-76$	$2.68E-75$	0	$1.48E-61$	$4.95E-63$	$2.71E-62$
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F4	BOA	$1.20E-11$	$1.68E-11$	$1.50E-11$	$9.51E-13$	$1.43E-11$	$1.75E-11$	$1.59E-11$	$8.93E-13$
	WOA	$4.59E-79$	$2.42E-66$	$8.32E-68$	$4.41E-67$	$2.65E-77$	$6.99E-66$	$2.47E-67$	$1.28E-66$
	SCA	$5.43E+03$	$3.79E+06$	$2.99E+05$	$6.84E+05$	$8.61E+08$	$8.53E+09$	$4.01E+09$	$1.68E+09$
	SSA	0	$1.01E-55$	$3.36E-57$	$1.84E-56$	$1.26E-248$	$1.51E-53$	$5.04E-55$	$2.76E-54$
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F5	BOA	$2.30E-12$	$1.40E-11$	$7.28E-12$	$2.44E-12$	$1.28E-11$	$1.63E-11$	$1.43E-11$	$7.15E-13$
	WOA	0	0	0	0	0	0	0	0
	SCA	$3.95E-01$	$1.65E+00$	$1.12E+00$	$2.96E-01$	$1.57E+01$	$7.96E+01$	$5.01E+01$	$1.73E+01$
	SSA	0	0	0	0	0	0	0	0
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F6	BOA	0	$1.34E-02$	$4.49E-05$	$2.45E-03$	0	0	0	0
	WOA	0	0	0	0	0	0	0	0
	SCA	$2.42E+02$	$4.29E+03$	$1.37E+03$	$1.02E+03$	$6.84E+04$	$3.47E+05$	$2.02E+05$	$8.54E+04$
	SSA	0	0	0	0	0	0	0	0
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F7	BOA	$2.55E-15$	$1.75E+01$	$1.89E+00$	$4.82E+00$	0	0	0	0
	WOA	0	$7.77E-01$	$3.95E-02$	$1.58E-01$	0	0	0	0
	SCA	$6.39E+00$	$1.77E+01$	$1.41E+01$	$2.59E+00$	$5.87E+01$	$2.34E+02$	$1.80E+02$	$5.13E+01$
	SSA	0	0	0	0	0	0	0	0
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F8	BOA	0	$3.59E+02$	$5.62E+01$	$1.28E+02$	0	0	0	0
	WOA	0	0	0	0	0	0	0	0
	SCA	$2.09E+02$	$2.27E+03$	$9.64E+02$	$5.04E+02$	$9.11E+04$	$3.56E+05$	$2.07E+05$	$6.99E+04$
	SSA	0	0	0	0	0	0	0	0
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F9	BOA	$4.88E+01$	$4.90E+01$	$4.89E+01$	$3.31E-02$	$4.99E+02$	$4.99E+02$	$4.99E+02$	$2.75E-02$
	WOA	$4.76E+01$	$4.87E+01$	$4.82E+01$	$3.93E-01$	$4.96E+02$	$4.98E+02$	$4.97E+02$	$3.84E-01$
	SCA	$2.37E+06$	$2.57E+09$	$5.79E+08$	$5.52E+08$	$1.12E+11$	$3.54E+11$	$2.42E+11$	$5.69E+10$
	SSA	$4.13E-07$	$8.29E-03$	$1.60E-03$	$2.17E-03$	$1.62E-06$	$7.75E-03$	$1.35E-03$	$1.92E-03$
	MRFO	$4.27E+01$	$4.48E+01$	$4.37E+01$	$5.71E-01$	$4.94E+02$	$4.97E+02$	$4.96E+02$	$6.72E-01$
	LGMRFO	$4.29E+01$	$4.44E+01$	$4.35E+01$	$2.82E-01$	$4.91E+02$	$4.92E+02$	$4.91E+02$	$1.97E-01$
F10	BOA	$9.12E-12$	$1.41E-11$	$1.17E-11$	$1.21E-12$	$1.04E-11$	$1.47E-11$	$1.28E-11$	$1.21E-12$
	WOA	$3.03E-89$	$1.82E-69$	$6.35E-71$	$3.33E-70$	$5.05E-83$	$5.29E-69$	$2.95E-70$	$1.04E-69$
	SCA	$8.10E+00$	$2.94E+03$	$8.03E+02$	$8.46E+02$	$3.40E+04$	$1.82E+05$	$1.08E+15$	$3.80E+04$

Table 3: Continued.

Function	Algorithm	$D = 50$				$D = 500$			
	SSA	0	$1.71E-35$	$5.70E-37$	$3.11E-36$	$4.61E-241$	$1.18E-54$	$3.94E-56$	$2.16E-55$
	MRFO	0	0	0	0	0	0	0	0
	LGMRFO	0	0	0	0	0	0	0	0
F11	BOA	$2.11E-10$	$4.35E-09$	$1.49E-09$	$1.11E-09$	$7.82E-11$	$2.23E-09$	$4.66E-10$	$4.12E-10$
	WOA	$8.88E-16$	$7.99E-15$	$3.85E-15$	$2.65E-15$	$8.88E-16$	$1.51E-14$	$5.03E-15$	$3.37E-15$
	SCA	$2.04E+01$	$2.06E+01$	$2.05E+01$	$6.67E-01$	$2.08E+01$	$2.09E+01$	$2.09E+01$	$2.84E-02$
	SSA	$8.88E-16$	$8.88E-16$	$8.88E-16$	0	$8.88E-16$	$8.88E-16$	$8.88E-16$	0
	MRFO	$8.88E-16$	$8.88E-16$	$8.88E-16$	0	$8.88E-16$	$8.88E-16$	$8.88E-16$	0
	LGMRFO	$8.88E-16$	$8.88E-16$	$8.88E-16$	0	$8.88E-16$	$8.88E-16$	$8.88E-16$	0
F12	BOA	$1.37E-03$	$1.01E-01$	$2.97E-02$	$2.70E-02$	$5.91E-06$	$3.51E-04$	$7.83E-05$	$6.69E-05$
	WOA	$1.52E-47$	$5.89E-01$	$1.18E-01$	$1.90E-01$	$4.32E-57$	$4.26E-13$	$1.42E-14$	$7.78E-14$
	SCA	$1.27E-01$	$7.06E-01$	$3.55E-01$	$1.33E-01$	$3.43E-02$	$1.39E-01$	$7.27E-02$	$3.02E-02$
	SSA	0	$3.85E-16$	$2.02E-17$	$7.22E-17$	0	$3.62E-16$	$2.15E-17$	$7.20E-17$
	MRFO	0	$2.57E-13$	$3.21E-14$	$4.70E-14$	0	$3.76E-11$	$4.21E-12$	$1.01E-11$
	LGMRFO	0	$4.30E-13$	$9.43E-14$	$1.27E-13$	0	0	0	0

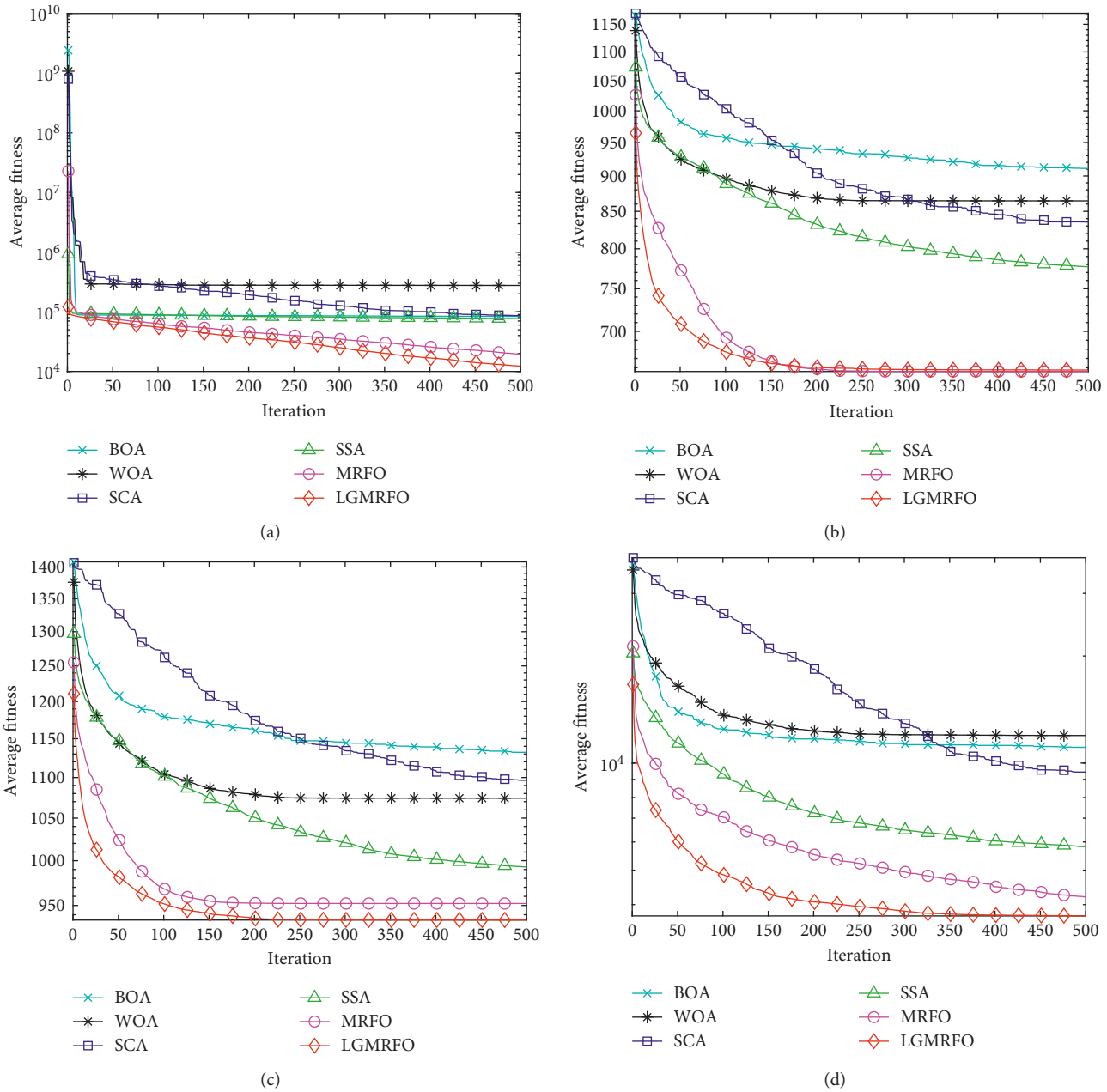
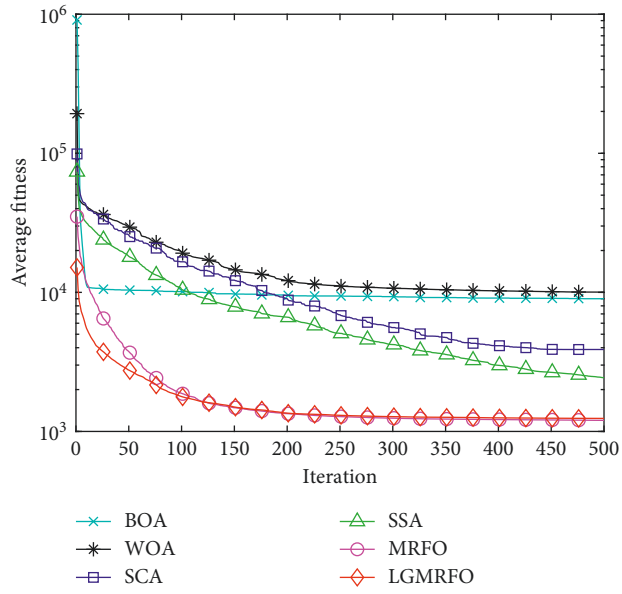
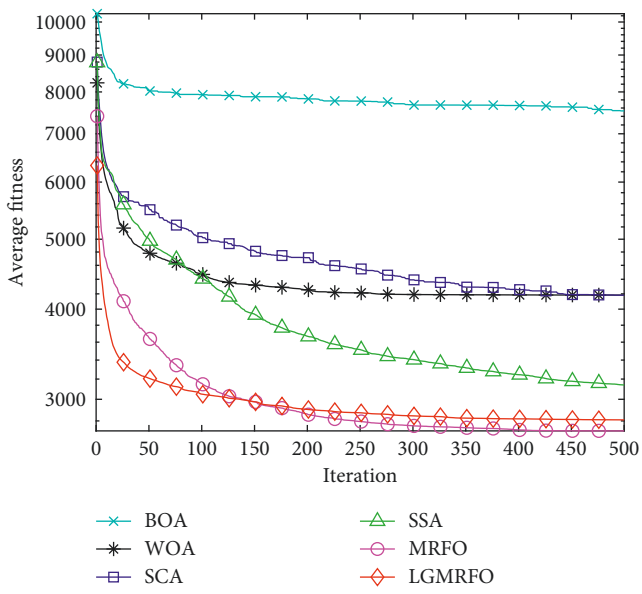


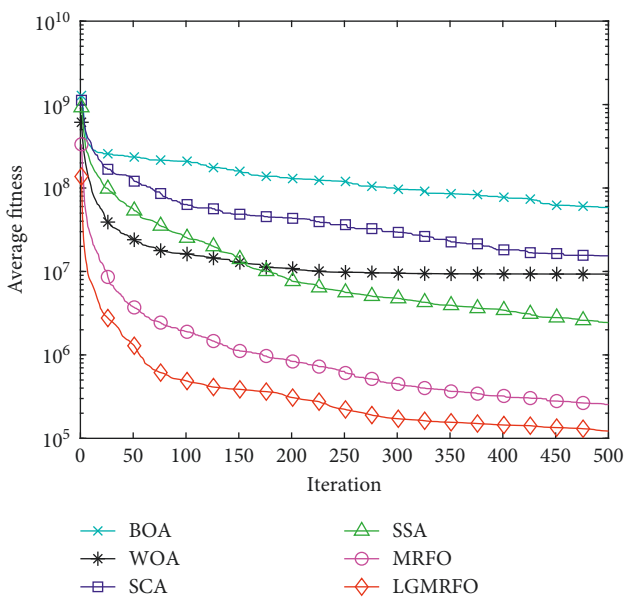
Figure 4: Continued.



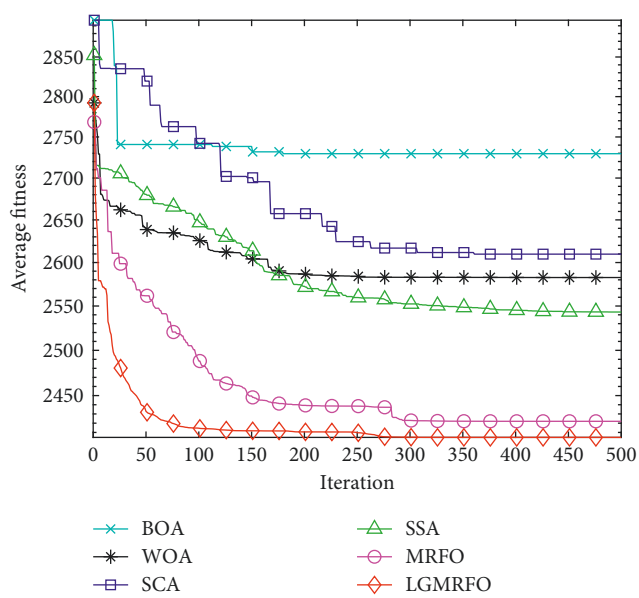
(e)



(f)

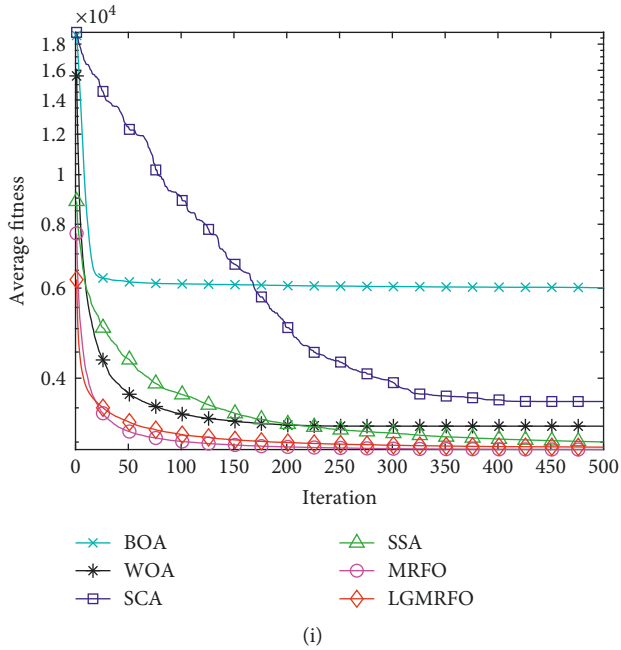


(g)



(h)

Figure 4: Continued.



(i)

Figure 4: Average convergence curves of some CEC2017 test suite functions. (a) CF2. (b) CF4. (c) CF7 (d) CF8. (e) CF10. (f) CF15. (g) CF17. (h) CF20. (i) CF24.

5.2.3. Evaluation of High-Dimensional Functions. Table 3 shows a comparison of the algorithms' outcomes in the 500-dimensional function test. It is obvious from a comparison of the experimental findings of the low-dimensional function test that LGMRFO gets better outcomes in terms of both search accuracy and convergence speed. The increase in dimensionality of the function from a low-dimensional to a high-dimensional function will affect the algorithm's convergence performance. Table 3 shows that both LGMRFO and basic MRFO approach the theoretical optimum with a standard deviation of 0 in the single peaked high-dimensional test function. This indicates that LGMRFO and MRFO are stable, and the convergence speed of LGMRFO is faster than other algorithms, as shown in Figures 3(a)~3(d), demonstrating the superiority of the improved strategy, whereas the convergence results of other compared algorithms are worse than the low-dimensional function. The standard deviation is also higher than in the low dimension, indicating that the other comparison algorithms are less robust on single-peaked high-dimensional functions; LGMRFO ranks first in the multipeaked high-dimensional function test, except for function F9; and LGMRFO ranks first in the low-dimensional multipeaked function F12, indicating that the improvement strategy in higher ability. In terms of convergence performance under high-dimensional functions, LGMRFO still outperforms the other five techniques.

5.3. Results Evaluation of CEC2017 Test Suite Functions. Table 4 shows a comparison of the algorithms' outcomes in some CEC2017 test suite functions. Figure 4 shows the average convergence curves of some CEC2017 test suite functions. Therefore, LGMRFO achieves the best results in CF2, CF4, CF7, CF8, CF10, CF17, and CF20. It shows that the overall performance of LGMRFO is powerful so that it can perform a smoother transition between exploration and exploration trends.

5.4. Wilcoxon Rank Sum Test. The Wilcoxon rank sum test [43] is a nonparametric statistical test that is performed to see if the LGMRFO method is significantly different from others. As a result, the results of the five algorithms were tested 30 times independently on 15 test functions and 9 CEC2017 functions as samples, and the Wilcoxon rank sum test was used to determine the significant difference between the solution results of the five compared algorithms and the LGMRFO solution results for the 50 and 500-dimensional, fixed-dimensional functions, and 9 CEC2017 functions, respectively. Tables 5–7 show the outcomes of the tests.

The null hypothesis is rejected when $P < 0.05$ indicates that the two algorithms are statistically different, whereas $P > 0.05$ implies that the two algorithms provide equivalent search results, according to the literature [44]. "NaN" implies that the associated algorithm searches for theoretical optimal solution, hence this hypothesis test is not applicable. In the 50-dimensional instance, the LGMRFO algorithm performs much better than the other examined algorithms, with the exception of MRFO, whereas in the 500-dimensional situation, the LGMRFO method performs significantly better than the 50-dimensional one. In the 9 CEC2017 functions situation, among the 45 data sets, 42 are less than 0.05, comprising 93.3% of the total data. This shows that LGMRFO has statistical advantages over the other

Table 4: Comparison of some CEC2017 test suite Functions.

Function	Algorithm	Best	Worst	Average	Std. Deviation
CF2	BOA	61865.0516	94673.9128	83610.9	7252.6901
	WOA	181153.9155	432212.7775	276488.167	55312.8565
	SCA	47819.1154	168455.467	86410.2938	22263.7286
	SSA	62613.8174	88822.6913	76844.6378	5701.0961
	MRFO	9829.1363	31305.175	19475.5722	5541.5783
	LGMRFO	**3696.8277**	**25977.9711**	**12196.0016**	**5512.4282**
CF4	BOA	877.3537	967.557	910.4024	22.3843
	WOA	729.3879	1020.0647	864.4385	73.0595
	SCA	786.6671	876.3863	835.3316	26.4226
	SSA	670.135	847.6568	777.11	42.704
	MRFO	598.501	749.7332	657.6086	38.2902
	LGMRFO	**596.5108**	**742.7687**	**656.0086**	**32.1436**
CF7	BOA	1105.4266	1162.1825	1132.1944	15.1434
	WOA	1009.6932	1229.7966	1074.2146	51.6852
	SCA	1065.2217	1126.971	1096.4516	17.2106
	SSA	891.4775	1049.7886	992.9483	34.9974
	MRFO	873.6269	1002.9701	952.1615	34.1025
	LGMRFO	**881.5865**	**982.2495**	**934.2627**	**22.9263**
CF8	BOA	8895.1653	12920.2913	11075.1726	1016.128
	WOA	6526.484	24593.4414	11953.6288	4545.5098
	SCA	5801.305	12951.4022	9435.4069	2185.7205
	SSA	5377.5911	6872.3515	5815.5711	314.8259
	MRFO	2164.7351	7093.8441	4205.0785	1036.6014
	LGMRFO	**2152.914**	**5618.2209**	**3717.7776**	**748.5012**
CF10	BOA	6547.2791	13835.6547	9002.8469	2183.7801
	WOA	3775.3694	17919.2339	10044.4225	3734.4153
	SCA	2566.8743	5716.6906	3883.7186	870.4679
	SSA	1492.0962	4381.4378	2434.9342	820.455
	MRFO	**1165.9774**	**1315.8124**	**1240.9612**	**47.1172**
	LGMRFO	1164.0194	1259.8653	1201.5794	37.8005
CF15	BOA	4532.5588	12202.6872	7530.0974	1860.0716
	WOA	3091.6252	6343.4343	4183.5431	668.3028
	SCA	3391.0211	4496.9845	4182.2468	265.6575
	SSA	2430.5504	4058.9642	3139.6938	393.3999
	MRFO	2095.1895	3166.5832	2712.2462	317.7892
	LGMRFO	**2229.9637**	**3421.9934**	**2809.234**	**277.3386**
CF17	BOA	4956069.9615	195429416.0647	58482274.1537	48599692.6692
	WOA	466476.5171	41365723.3653	9301704.5173	10116889.255
	SCA	2977781.3915	38837593.3164	15400703.9766	9893645.6625
	SSA	81791.5292	12803104.8522	2438302.354	2728490.4502
	MRFO	41716.4756	962707.3555	254339.6966	203873.0355
	LGMRFO	**28304.0738**	**430158.1843**	**122081.5603**	**102567.4144**
CF20	BOA	2612.4543	2931.4312	2729.5077	205.2435
	WOA	2412.3452	2734.6753	2582.6538	66.6743
	SCA	2400.7732	2714.5564	2609.7752	76.453
	SSA	2423.4533	2612.1334	2543.1367	23.8764
	MRFO	2201.1145	2511.1134	2422.0052	16.0254
	LGMRFO	**2301.7768**	**2501.657**	**2404.667**	**13.1909**
CF24	BOA	4867.6438	7550.4929	6012.4665	605.8525
	WOA	3111.795	3356.4394	3222.9168	72.9402
	SCA	3305.9347	4349.4399	3601.44	268.54
	SSA	2937.7087	3081.5462	3001.3027	38.5158
	MRFO	**2885.7049**	**2942.7045**	**2900.6972**	**17.214**
	LGMRFO	2891.818	2979.9971	2932.3753	24.3826

Table 5: Wilcoxon rank sum test results for 50 and fixed dimensions.

F	MRFO	SSA	SCA	WOA	BOA
F1	NaN	$1.21E-12$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F2	NaN	$1.21E-12$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F3	NaN	$1.66E-11$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F4	NaN	$1.93E-10$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F5	NaN	NaN	$1.21E-12$	0.3337	$1.21E-12$
F6	NaN	NaN	$1.21E-12$	NaN	0.0013
F7	NaN	NaN	$1.21E-12$	NaN	$1.21E-12$
F8	NaN	NaN	$1.21E-12$	0.3337	$1.95E-09$
F9	0.0023	$3.02E-11$	$3.02E-11$	$3.02E-11$	$3.02E-11$
F10	NaN	$1.66E-11$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F11	NaN	NaN	$1.21E-12$	$2.17E-07$	$1.21E-12$
F12	0.8877	0.2010	$2.40E-11$	$2.42E-07$	$2.40E-11$
F13	0.0419	$3.46E-10$	$4.28E-11$	$3.86E-11$	$5.84E-11$
F14	0.0080	$4.09E-11$	$1.57E-11$	$1.57E-11$	$1.57E-11$
F15	0.3128	$3.02E-06$	$1.72E-12$	$1.72E-12$	$1.72E-12$

Table 6: Wilcoxon rank sum test results for 500 dimensions.

F	MRFO	SSA	SCA	WOA	BOA
F1	NaN	$5.77E-11$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F2	0.3337	$5.77E-11$	$1.69E-14$	$1.69E-14$	$1.69E-14$
F3	NaN	$4.57E-12$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F4	NaN	$4.57E-12$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F5	NaN	NaN	$1.21E-12$	NaN	$1.21E-12$
F6	NaN	NaN	$1.21E-12$	NaN	NaN
F7	NaN	NaN	$1.21E-12$	NaN	NaN
F8	NaN	NaN	$1.21E-12$	NaN	0.3337
F9	$3.02E-11$	$3.02E-11$	$3.02E-11$	$3.02E-11$	$3.02E-11$
F10	NaN	$4.57E-12$	$1.21E-12$	$1.21E-12$	$1.21E-12$
F11	NaN	NaN	$1.21E-12$	$3.66E-08$	$1.21E-12$
F12	$1.27E-05$	$1.21E-12$	$1.21E-12$	$1.21E-12$	$1.21E-12$

Table 7: Wilcoxon rank sum test results for 9 CEC2017 functions.

F	MRFO	SSA	SCA	WOA	BOA
CF2	$1.5292E-05$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$
CF4	0.9	$4.1997E-10$	$3.0199E-11$	$4.0772E-11$	$3.0199E-11$
CF7	0.040595	$1.5581E-08$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$
CF8	0.096263	$5.4941E-11$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$
CF10	0.00047138	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$
CF15	0.26433	0.0015178	$3.6897E-11$	$8.1527E-11$	$3.0199E-11$
CF17	0.00065486	$3.8249E-09$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$
CF20	$6.5991E-07$	$3.8249E-09$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$
CF24	$7.5991E-07$	$3.8249E-09$	$3.0199E-11$	$3.0199E-11$	$3.0199E-11$

Table 8: Parameter setting for WSN coverage.

Parameters	Values
Region	$50\,m \times 50\,m$
Number of nodes	30/35
Perceived radius	5 m
Communication radius	10 m

competitive algorithms. In conclusion, LGMRFO outperforms MRFO, SSA, SCA, WOA, and BOA by a statistically significant margin, indicating that the LGMRFO algorithm is statistically superior.

6. Coverage Optimization of WSN Based on LGMRFO

6.1. WSN Node Coverage Model. The Boolean measurement model and the probabilistic measurement model are the two basic types of WSN node coverage models [45]. In this research, we calculate network coverage using the more standard Boolean model.

Assume that in a square WSN monitoring region with a side length of L, N isomorphic sensor nodes are randomly distributed. Assume that the set of nodes is $V = \{v_1, v_2, \ldots, v_N\}$, with node v_i's location coordinates being

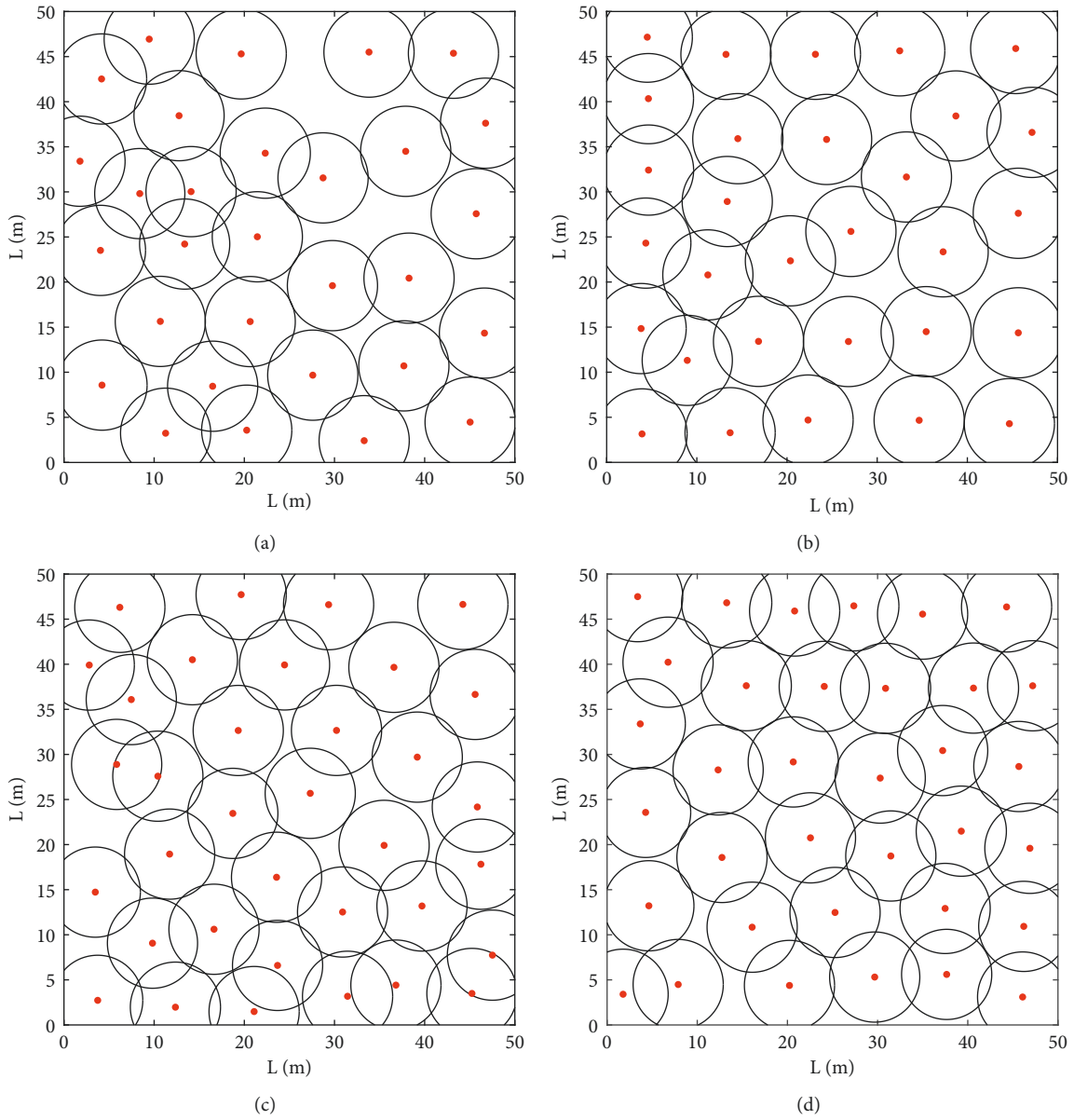


Figure 5: Node distribution before and after algorithm optimization: (a) Coverage result of MRFO ($N = 30$), (b) Coverage result of LGMRFO ($N = 30$), (c) Coverage result of MRFO ($N = 35$), and (d) Coverage result of LGMRFO ($N = 35$).

(x_i, y_i), and each node's sensing radius being R_s. The area is discretized into $m \times n$ target grid points to be covered to make the calculation easier, and the set of target points is indicated as $u_j = (x_j, y_j), j \in \{1, 2, \ldots, m \times n\}$. The distance between the sensor node and the target point is specified as

$$d(v_i, u_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{16}$$

The target point has been covered if there is a node whose distance from the target point is less than or equal to the sensing radius R_s. According to the Boolean model, the chance that the sensor node v_i detects the target location is defined as

$$p(v_i, u_j) = \begin{cases} 0, & d(v_i, u_j) > R_s, \\ 1, & d(v_i, u_j) \le R_s. \end{cases} \tag{17}$$

When the target point is sensed by more than one sensor, the joint sensing probability of the target point is defined as

$$p(\mathbf{V}, u_j) = 1 - \prod_{i=1}^{N} [1 - p(v_i, u_j)]. \tag{18}$$

The area network coverage is calculated by multiplying the sum of the total perceived probability of target points covered by a set of nodes by the entire number of target points in the area.

Table 9: Average coverage.

Algorithm	Average coverage/%			
	30 nodes	30 nodes initialization	35 nodes	35 nodes initialization
MRFO	82.79	63.36	89.24	67.47
LGMRFO	83.87	63.65	90.66	68.89

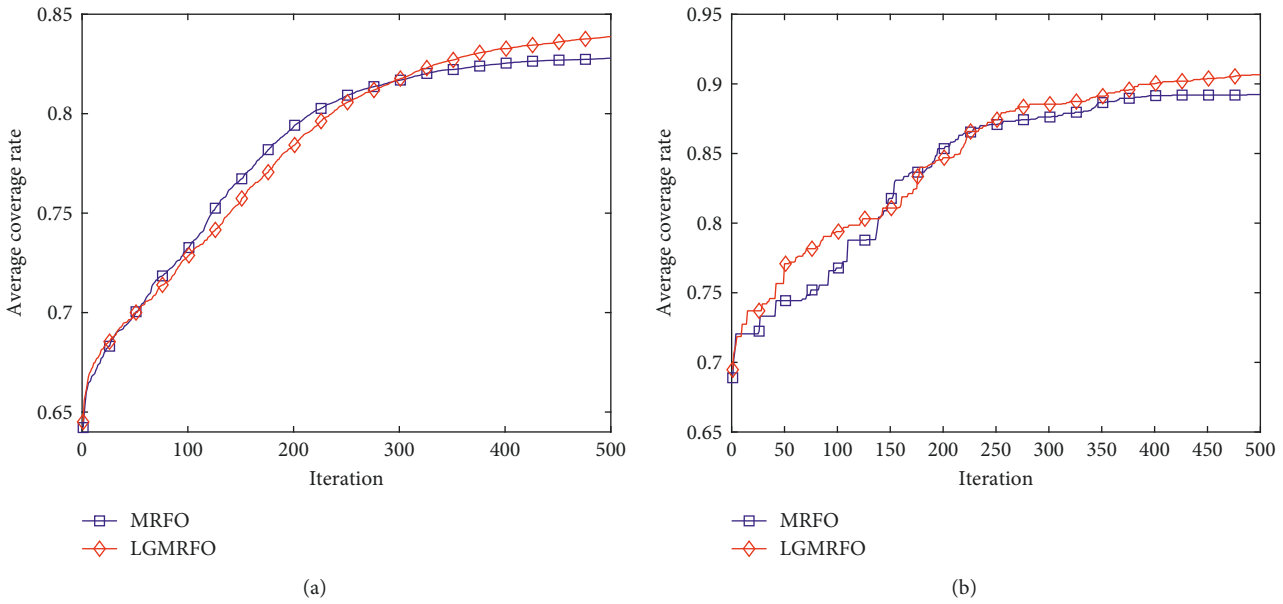


Figure 6: Average coverage iteration curve: (a) Average coverage iteration curve ($N = 30$) and (b) Average coverage iteration curve ($N = 35$).

$$R_{\text{cov}} = \frac{\sum_{j=1}^{m\times n} p\left(V, u_j\right)}{m \times n}. \tag{19}$$

As a result, the WSN coverage optimization issue can be defined as the coverage of complete target grid points by N sensor nodes on the monitoring area using an optimization technique, which can then be turned into a single objective optimization problem that maximizes equation. (17), i.e.,

$$\max\left(\frac{\sum_{j=1}^{m\times n} p\left(V, u_j\right)}{m \times n}\right). \tag{20}$$

6.2. Analysis of Simulation. Two sets of experiments are used in this work to verify the efficiency of LGMRFO on WSN coverage optimization. As stated in Table 8, the experimental settings have been set.

Figure 5 shows the results of sensor area coverage after algorithm optimization. The distribution at 30 nodes is shown in Figure 5(a) and 5(b), with MRFO covering 82.43 percent of the nodes and LGMRFO covering 84.78 percent. In the monitoring region, there are still coverage blind spots, and node overlapping coverage is more evident, as shown in Figure 5(a), but the optimized nodes in Figure 5(b) are more uniformly distributed. Figures 5(c) and 5(d) show the coverage results when 35 nodes are installed. After MRFO optimization, the coverage rate is 89.43%, yet there are coverage blind patches near the monitoring area's edge. After LGMRFO optimization, the coverage rate is 92.62%, and the node overlapping area is greatly reduced.

Table 9 shows the coverage of MRFO and LGMRFO running independently for 20 times and each operation iteration for 500 times, respectively. As can be seen from Table 9, LGMRFO's final and initial coverage are higher than those of the MRFO algorithm, indicating that the LHS method's enhanced strategy and location update improve the algorithm's search accuracy.

The average coverage iteration curves are given in Figure 6. LGMRFO coverage in the middle of iteration is slightly lower than MRFO at 30 nodes in Figure 6(a), which is owing to the premature maturity produced by MRFO converging too quickly. LGMRFO gradually surpasses MRFO after 300 iterations, suggesting that MRFO has entered the local optimum, whereas LGMRFO jumps out of the local optimum and optimization accuracy improves, demonstrating that the group learning technique is effective. The population's health (node distribution) has improved, and the coverage rate has continuously increased. LGMRFO's coverage is greater than MRFO's when 35 nodes are deployed, which corresponds to an increase in individual dimension, and both the convergence speed and coverage are much greater than the MRFO algorithm's average optimization result.

In summary, by comparing the experimental results of deploying different numbers of nodes, LGMRFO achieves

higher average network coverage under the same conditions, and the node layout is more reasonable, resulting in fewer coverage blind areas and overlapping areas, proving the effectiveness of the improved strategy.

7. Conclusion

To overcome the inadequacies of the manta ray foraging optimization method in terms of optimization accuracy, this work offers an improved manta ray foraging optimization algorithm (LGMRFO). Firstly, to improve the quality of the initial population, the LHS method is used to homogenize the population position distribution. Secondly, the Levy flight and adaptive t-distribution variation strategies are used before the cyclone foraging exploration phase and somersault foraging behavior, respectively, so as to improve the algorithm's ability to jump out of the local optimum. Finally, a group learning strategy is used for the updated population. On 24 typical test functions, the LGMRFO algorithm is compared to the other five algorithms, and the method significance level is validated using the Wilcoxon rank sum test. LGMRFO greatly enhances convergence speed, optimization-seeking accuracy, and global optimization capability, according to the findings of the experiments. Finally, on the WSN coverage optimization problem, LGMRFO is compared to MRFO, and the experimental findings support the usefulness of the proposed improvement strategies.

As future challenges, different applications other than WSN coverage optimization of LGMRFO can be explored and its capabilities in dealing with difficult test problems can be examined. Besides, new configurations of this algorithm can be considered as other researchers may have different viewpoints on the presented methodology.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. E. N. G. Peng, Z. Ni, and X. Zhu, "Attribute reduction method based on improved binary glowworm swarm optimization algorithm and neighborhood rough set," *Pattern Recognition and Artificial Intelligence*, vol. 33, no. 2, pp. 95–105, 2020.

[2] X. Zhou, Y. Liu, and X. U. Chong-chong, "A hybrid state transition optimization algorithm based on adaptive quasi-Newton method and its application," *Control and Decision*, vol. 36, no. 10, pp. 2451–2458, 2021.

[3] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Applied Soft Computing*, vol. 76, pp. 16–30, 2019.

[4] S. Ray, A. Das, K. G. Dhal, J. Gálvez, and P. K. Naskar, "Cauchy with whale optimizer based eagle strategy for multi-level color hematology image segmentation," *Neural Computing & Applications*, vol. 33, no. 11, pp. 5917–5949, 2021.

[5] A. K. Barisal and S. Mishra, "Improved PSO based automatic generation control of multi-source nonlinear power systems interconnected by AC/DC links," *Cogent Engineering*, vol. 5, no. 1, Article ID 1422228, 2018.

[6] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.

[7] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[8] S. C. A. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *KnowledgE – Based Systems*, vol. 96, pp. 120–133, 2016.

[9] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[10] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a natur*E*– inspired metaheuristic," *Expert Systems with Applications*, vol. 152, Article ID 113377, 2020.

[11] B. Abdollahzadeh, F. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: a new nature inspired metaheuristic algorithm for global optimization problems," *Computers & Industrial Engineering*, vol. 158, Article ID 107408, 2021.

[12] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, Article ID 103300, 2020.

[13] D. Izci, S. Ekinci, E. Eker, and M. Kayri, "Improved manta ray foraging optimization using opposition-based learning for optimization problems," in *Proceedings of the International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–6, IEEE, Ankara, Turkey, June 2020.

[14] B. Sheng, T. Pan, Y. Luo, and K. Jermsittiparsert, "System identification of the PEMFCs based on balanced manta-ray foraging optimization algorithm," *Energy Reports*, vol. 6, pp. 2887–2896, 2020.

[15] O. E. Turgut, "A novel chaotic manta-ray foraging optimization algorithm for thermo-economic design optimization of an air-fin cooler," *SN Applied Sciences*, vol. 3, p. 3, 2021.

[16] F. Fausto, A. Reyna-Orta, E. Cuevas, Á. G. Andrade, and M. Perez-Cisneros, "From ants to whales: metaheuristics for all tastes," *Artificial Intelligence Review*, vol. 53, no. 1, pp. 753–810, 2020.

[17] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.

[18] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: a novel optimization algorithm," *KnowledgE – Based Systems*, vol. 191, Article ID 105190, 2020.

[19] M. Azizi, "Atomic orbital search: a novel metaheuristic algorithm," *Applied Mathematical Modelling*, vol. 93, pp. 657–683, 2021.

[20] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Transient search optimization: a new meta-heuristic optimization algorithm," *Applied Intelligence*, vol. 50, no. 11, pp. 3926–3941, 2020.

[21] I. Ahmadianfar, A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "Run beyond the metaphor: an efficient

optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, Article ID 115079, 2021.

[22] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: a new metaheuristic optimization algorithm," *Information Sciences*, vol. 540, pp. 131–159, 2020.

[23] T. Erkan, D. Gokhan, and G. Sine, "A novel math-inspired algorithm," *Advances in Electrical and Computer Engineering*, vol. 17, p. 2, 2017.

[24] L. Abualigah, D. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.

[25] I. Ahmadianfar, A. Heidari, S. Noshadian, H. Chen, and A. H. Gandomi, "INFO: an efficient optimization algorithm based on weighted mean of vectors," *Expert Systems with Applications*, vol. 195, Article ID 116516, 2022.

[26] Q. Askari, Y. Younas, and M. Saeed, "Political optimizer: a novel socio-inspired meta-heuristic for global optimization," *KnowledgE – Based Systems*, vol. 195, Article ID 105709, 2020.

[27] Y. Zhang and A. Chi, "Group teaching optimization algorithm with information sharing for numerical optimization and engineering optimization," *Journal of Intelligent Manufacturing*, 2021, https://doi.org/10.1007/s10845-021-01872-2.

[28] J. Zhang, M. Xiao, L. Gao, and Q. Pan, "Queuing search algorithm: a novel metaheuristic algorithm for solving engineering optimization problems," *Applied Mathematical Modelling*, vol. 63, pp. 464–490, 2018.

[29] B. Das, V. Mukherjee, and D. Das, "Student psychology based optimization algorithm: a new population based optimization algorithm for solving optimization problems," *Advances in Engineering Software*, vol. 146, Article ID 102804, 2020.

[30] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[31] Y. Yang, H. Chen, A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.

[32] A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[33] Y. S. Mohamed, A. A. M. Mohamed, A. A. M. El-Gaafary, and A. M. Hemeida, "Optimal power flow using moth swarm algorithm," *Electric Power Systems Research*, vol. 142, pp. 190–206, 2017.

[34] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2015.

[35] A. Mohammadi-Balani, M. D. Dehghan Nayeri, A. Azar, and M. Taghizadeh-Yazdi, "Golden eagle optimizer: a natur*E*-– inspired metaheuristic algorithm," *Computers & Industrial Engineering*, vol. 152, Article ID 107050, 2021.

[36] L. Xie, T. Han, and H. Zhou, "Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9210050, 22 pages, 2021.

[37] C. Chen, J. Wen, and S. Cheng, "Probabilistic load flow method based on nataf transformation and Latin hypercube sampling," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 2, pp. 294–301, 2013.

[38] G. Iacca, S. J. Vcd, and V. Melo, "An improved Jaya optimization algorithm with Lévy flight," *Expert Systems with Applications*, vol. 165, no. 1, Article ID 113902, 2020.

[39] Q. Li, J. Chen, and F. Pan, "Robust PPLS regression modeling subject to t-distributed noise," *Systems Engineering-Theory & Practice*, vol. 38, no. 9, pp. 2416–2423, 2018.

[40] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[41] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[42] N. H. Awad, M. Z. Ali, and J. J. Liang, "Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization," *Technical Report*, https://www.researchgate.net/publication/317228117_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2017_Competition_and_Special_Session_on_Constrained_Single_Objective_Real-Parameter_Optimization, 2016.

[43] H. Zhao, L. Mu-dong, and W. E. N. G. Xing-wei, "Performance evaluation for biology-inspired optimization algorithms based on nonparametric statistics," *Journal of Air Force Engineering University (Natural Science Edition)*, vol. 16, no. 1, pp. 89–94, 2015.

[44] J. Derrac, A. S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[45] F. Zhu and W. Wang, "A coverage optimization method for WSNs based on the improved weed algorithm," *Sensors*, vol. 21, no. 17, p. 5869, 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6925668, 16 pages
https://doi.org/10.1155/2022/6925668

Research Article

Personalized Hybrid Education Framework Based on Neuroevolution Methodologies

Wenjing Yin

Zhengzhou Preschool Education College, Zhengzhou, 450000, China

Correspondence should be addressed to Wenjing Yin; wenjing_yin66@126.com

Received 16 February 2022; Revised 8 April 2022; Accepted 15 April 2022; Published 19 May 2022

Academic Editor: Diego Oliva

The future pedagogical systems need anthropocentric inclusive educational programs in which the goal should be adjustable according to the knowledge requirements, intelligence, and learning objective of each student. Prioritizing these needs, innovative AI methods are required to assist and ensure the making of conscious educational decisions, in terms of clear identification and categorization with high accuracy of various forms of skills and knowledge of each student. This paper proposes a neuroevolution emerging technique that combines the searchability of evolutionary computation and the learning capability of a hybrid artificial neural networks method. Specifically, the proposed growing semiorganizing neural gas (GsONG) is a practical AI methodology utilizing advanced clustering techniques to enhance the learning experience by categorizing the true abilities, skills, and needs of learners, in an inclusive differentiated learning framework. It is a neural network architecture that includes competing and cooperating neurons with an unstructured mode whereby a cooperation-competition process delimits the topological neighborhood of neurons in a grid to identify patterns for which their classes are not known. To optimize the above process, a heuristic method was used that investigates the space of an objective function by regulating the optimal topologies of neurons that form pathway segments in a semi-contemplative manner. Based on the extensive experiments and results obtained from the GsONG clustering approach, the proposed algorithm can compensate with high accuracy for difficulties in multicriteria grouping and differentiation of uncertainty structures such as in small or tiny data sets.

1. Introduction

Inclusive education as a part of differentiated teaching is an organized strategy that is a product of interdisciplinary collaboration and that as a pedagogical approach places at the center the uniqueness of each student, their unique skills, and their needs [1, 2]. It is concerned with the fact that teaching and learning should start from the level of students, instead of being based on a specific and predetermined action plan, which does not include students' readiness, interest, and learning profile [3].

It is essentially the organizational and pedagogical adaptation of inclusive education to meet the various needs of students in mixed grades (lean, mediocre, more capable, and charismatic students) while taking into account the differences of students in terms of readiness, learning style, and interests. In a learning environment characterized by student diversity, each student's personal abilities and strengths are recognized, and he is given opportunities to utilize and demonstrate his various skills through a variety of assessment techniques. It is important to say that differentiated teaching in a learning environment where there is a diversity of students is properly organized and planned and in no way is a product of a random or unbalanced process of allocating learning resources [4].

In this sense, inclusive differentiated teaching is more qualitative than quantitative, with the emphasis on the fact that some students are not given more work than others but that the work is tailored to their specific needs. In this context, interconnected and adequately planned educational activities are offered based on the uniqueness of each student, their skills, and their level. In this way, it is possible to have multiple approaches to the content, the course, and the result of the educational process.

As appealing as teaching differentiation is, it is true that in order for this differentiation to occur and work in a classroom, the curriculum must be adapted. Achieving it is directly related to the functional ways in which students' critical thinking development can take place, as well as through the opportunities offered to students to demonstrate what they have learned [5]. In conclusion, effective differentiated teaching utilizes various methods, means, and materials manage to meet the special needs of all students by increasing their learning opportunities, offering a comfortable and positive environment, where all students achieve high academic achievements [6].

The aim of this paper is to propose a technological system that utilizes computer intelligence algorithms to facilitate decision-making related to curriculum adaptation and categorization of students, based on individual assessments of their unique characteristics. This methodology provides the technological background for automated differentiation of teaching, taking into account characteristics such as the treatment of each student as a person with a history of learning in and out of school, modeling of student metacognitive development, linking to prior knowledge, and building knowledge with gradual progress, according to the perceiving level of each student [7]. Specifically, a differentiated learning framework is proposed, which, with the extensive use of machine learning algorithms, as well as optimization methods, creates a clear framework for multicriteria assessments, to classify students into small groups. The students belonging to each group will have uniform features of learning ability, difficulties in the cognitive object, and their psychosocial and perceptive summary.

The objective of the learning activity determines whether to form groups with students of comparable or mixed ability. If the goal of a group learning activity is to assist struggling pupils, research reveals that heterogeneous groups are the most effective. On the other side, homogeneous grouping might be preferable if the goal is to promote medium-ability groups to learn at high levels. In an educational setting, homogeneous grouping is defined as grouping students of similar instructional levels together so that they can work on materials that are best suited to their individual strengths and areas for improvement. Assessment and instructor observation are frequently used to identify these ability levels [8]. Homogeneous groups enable teachers to create lesson plans that are suited to their students' skills and save time by addressing individual requirements. In any event, because student ability levels differ by subject, categorizing students based on their skills ensures that they are ranked appropriately even within homogeneous groups [7].

On the other hand, heterogeneous groups are preferable if the goal is to raise difficult pupils and help them develop the independence and collaborative skills that come with reduced reliance on the teacher. However, this technique will reduce the value of the exercise for talented and at-grade pupils, even if they will have the opportunity to practice communication and leadership abilities. While some bright children enjoy the benefits and responsibilities that varied groups offer, others dislike them. Knowing the hierarchy of competency and talents, as well as what their peers are capable of, such students are aware that they may be responsible for a bigger portion of the work while dealing with apathetic or disruptive group members. Above-grade-level pupils are unlikely to be challenged by an exercise and will instead become annoyed by everyone's inability to accomplish it as well or as quickly as they could. While they will have the opportunity to develop other talents, they will miss out on the opportunity to learn as much as possible from the task at hand. In the worst-case situation, if a student is very spirited or requires regular stimulation, they may become disruptive and obstruct the learning of the rest of the group or perhaps the entire class. There is a danger that pupils in grade level and below will take a back seat in heterogeneous groups and allow their more gifted classmates to do all the work. This is especially true if the more capable students have assertive personalities, a strong desire to complete the assignment as quickly as possible, and a dismissive attitude toward their peers' talents and efforts.

The proposed system can be used to implement individualized teaching programs in homogeneous learning groups, based on the diversity that characterizes each student and person in general.

2. Related Literature

The related literature of the proposals on the educational methods to provide differentiated and personalized training mainly focuses on theoretical proposals. The potential of being able to provide personalized training in the educational process is great because it gives to the learner the best possible learning experience. Personalized e-learning encompasses a variety of educational technology and pedagogical methods that take into account individual student variations [4] and can customize the generic virtual training environment to meet their own needs. The competence of professors and educational material quality are key factors in e-learning programs [9], but instructors' competency is revealed through their approach, teaching style, and assistance throughout the entire online education experience. All these elements contribute to the positive influence on students' emotions and subsequently trigger flow. Ardura and Artola [10] intend to contribute to the flow and fill the hole that has been identified in terms of knowledge possessed in personalized e-learning settings in their extensive investigation. They also look at how education-related and subjective factors interact to create flow in a personalized e-learning environment, as well as evidence supporting the moderating influence of individual differences in gender and academic achievement. Even though this is detailed research, the limitations are that the timeframe for the calculations is limited making it impossible to say if the reported associations will alter over time as new e-learning methods for customization emerge. Furthermore, the sample of the study is limited to just one online college leaving the field open for contribution and enhancement of this model and the generalization of the results through further study.

In his paper, Nganji [11] provides a concept for a learning environment that allows learners to create their own online learning spaces and engage with pooled

materials. This model is called Flexible and Accessible User Constructed Learning Environment (FAUCLE) and essentially is a learner-centered model that connects the elements that contribute to the e-learning process and creates relationships among them. These are the learner, the educator, the accessibility expert, the e-learning programmer, the learning environment, and the learning content apps. The prerequisite for this model to take place is for the institution that provides the training to make applications and other resources available to these students in a form that is both accessible and useable, allowing them to create their learning environments with flexibility. The fact that this model is theoretical means that it needs to be implemented and tested to yield empirical data that can be compared to other studies. Milicevich and Ivanovic [7] in their work aim to address the research by examining aspects of tailored e-learning, as well as intelligent and interactive technologies. They show online education systems that are currently state-of-the-art that are powered by artificial intelligence at the end. Their study is also theoretical and serves as a suggestion for institutions and organizations who want to adopt these new technologies and approaches in e-learning. Maghsudi et al. [12] in their study give a quick rundown of current research, look into the challenges of AI/ML-based individualized education, and propose possible solutions. They conclude that "personalized education" is one of artificial intelligence's most valuable educational merits because it significantly improves education quality in several dimensions by adapting to the unique characteristics and expectations of each learner, such as personality, talent, objectives, and background. In addition, in unusual circumstances such as the COVID-19 outbreak [13] or natural disasters, online teaching is invaluable. Indeed, traditional education requires substantially more resources than online education in terms of classroom space, scheduling, and human resources, making it vulnerable to failure in the event of even minor changes in circumstances. As a result, new alternatives are unavoidable [14, 15]. Personalized education, despite the potential for a dramatic shift from traditional to modern education paradigms, is fraught with difficulties.

From the above literature, the conclusion is that the research community focuses on finding ways to combine technology with the traditional educational processes to provide a differentiated experience. What this paper does is go a step further and propose a practical methodology utilizing machine learning to enhance the learning experience.

3. The Gsong

To create an intelligent framework for categorizing the true abilities, skills, and needs of learners, a differentiated learning framework is proposed, which makes extensive use of nonsupervised machine learning technologies to achieve its goals [16]. Unsupervised learning is the ability to identify patterns for which their classes are not known, and the system generates predictions based on some distribution or some quantitative measures to evaluate and characterize the similarity of data in corresponding groups called clusters. The general clustering technique [17] is based on the idea that a data set $D = \{x_1, x_2, \ldots, x_n\}F$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a characteristic of the set $X \in R^r$ and r is the number of dimensions in data. After defining the set of blocks K, assign each point $x^{(i)}$ of the data set to a block C_k so that the $S_{core}(C, D)$ is maximized or minimized as appropriate. How to calculate a function of this type is given by the following formula [17]:

$$S_{\text{core}}(C, D) = \sum_{k=1}^{K} d(x, c_k), \tag{1}$$

where $c_k = 1/n_k \sum_{x \in c_k} x$ and $d(x, y) = \|x - y\|^2$.

Elements that belong to a cluster show greater or relatively greater similarity. The training of a computational intelligence model with the method of clustering is called to calculate and finally classify into clusters, data $x_1, \ldots, x_n$ without giving the values $f(x_1), \ldots, f(x_n)$. In this work, the proposed model applies clustering based on competitive learning and is specifically implemented using artificial neural networks [14, 18].

Specifically, the methodology provides for the classification of students into homogeneous groups based on their particular characteristics [19]. For this reason, a neural network was created that includes a competitive layer of competing neurons. Each competing neuron i is characterized by a weight vector $w_i = (w_{i\,d}, \ldots, w_{i\,d})^T$, $i = 1, \ldots, M$, and computes a similarity measure among the input information $x_i = (x_{i\,d}, \ldots, x_{i\,d})^T$ $x \in R$ and in w_i weights vector. Each time a student's characteristics appear as an entry in the network, competition is created between competing-level neurons to identify the winning neuron whose weight vector shows the greatest similarity to that input. The winning neuron m sets the output of $o_m = 1$, while the other neurons give $o_i = 0$, where $i = 1, \ldots, M$ and $i \neq m$. As a measure of similarity to find the winner neuron, a function inversely proportional to the Euclidean distance $\|x - w_i\|$ of the input vector x^n from the vector of w_i weights was used. Therefore the proposed neural network implements a representation of the input x, dimension d, in the coordinates of the grid $r_m = (z_{m1}, z_{m2})^T$.

More specifically, the proposed neural system forms a self-organizing map of its structures, starting from the process of initializing the weights $w_i = (w_{il}, \ldots, w_{i\,d})^T$. To achieve this, small weight values generated by a random number generator are given. The weight table of the hidden layer H is calculated as follows [20]:

$$\mathrm{H} = g(\omega x + b). \tag{2}$$

The output weights β are calculated based on the following function:

$$\beta = \left(\frac{\mathrm{I}}{C} + H^T H\right)^{-1} H^T X, \tag{3}$$

where $H = [h_1, \ldots, h_N]$ is the hidden level outputs and $X = [x_1, \ldots, x_N]$ is the input data. β can also be calculated from the general relation as follows:

$$\beta = \mathrm{H}^{+}T. \tag{4}$$

After this initialization, there are three basic procedures:

Competition. For each x^n, the grid neurons calculate the similarity function. The winner is the neuron with the highest similarity value. The Euclidean distance between $x = (x_1, \ldots, x_d)^T \; x \in R$ and $w_i = (w_{il}, \ldots, w_{i\,d})^T$ of the opposing neurons is used as a function of similarity.

Cooperation. The winner-neuron delimits the topological neighborhood of neurons in the grid, which will adjust their weights to the input vector. $h_{j,i}$ denotes the topological neighborhood centered on the winner-neuron i, which includes a set of neurons, one of which is denoted as j. Also denoted by $d_{j,i}$ is the distance between the winner neuron I and a neuron j. Thus, the topological neighborhood satisfies the above constraints [17, 21].

$$h_{j,i}(x) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right), \tag{5}$$

where parameter σ is the topological neighborhood's effective width, which defines how many neurons in the winner's neighborhood participate in the training process. This parameter decreases in each season n at an exponential rate according to the relation [21, 22]:

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right), \; n = 0, 1, 2, \ldots, \tag{6}$$

where parameter σ_0 is the initial value of the active amplitude and τ_1 is the polarity constant of the network.

Synaptic Adaption. In this last stage of the training process, the weights of the competing-level neurons are updated. The amount of this change is given by the following relation [23, 24]:

$$\Delta w_j = \eta h_{j,i}(x)\left(x - w_j\right), \tag{7}$$

where i is the winning neuron and j is the neuron in the neighborhood of i. Finally, given the vector of weights $w_j(n)$ for a given time n, the new vector for the time $n + 1$ can be calculated from the following relation:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i}(x)(n)\left(x(n) - w_j(n)\right). \tag{8}$$

From the above relation, it follows that the learning rate $\eta(n)$ depends on time. More specifically, it starts from an initial value of η_0 and decreases exponentially with increasing time n [25]:

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right), \; n = 0, 1, 2, \ldots, \tag{9}$$

where τ_2 is the polarization constant of the network.

In addition, the above process is divided into two phases:

Ordering phase is the initial phase, and it is during this phase that the competing-level weights are topologically arranged. During this phase, the learning rate η *(n)* starts from a value around 0.1 and gradually decreases, up to the value of 0.01. These values are achieved as follows:

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_1}\right), \; n = 0, 1, 2, \ldots, \quad \text{with } \eta_0 = 0.1 \text{ and } \tau_2 = 1000. \tag{10}$$

The following network polarization values were used to calculate the learning rate and active amplitude values in each iteration:

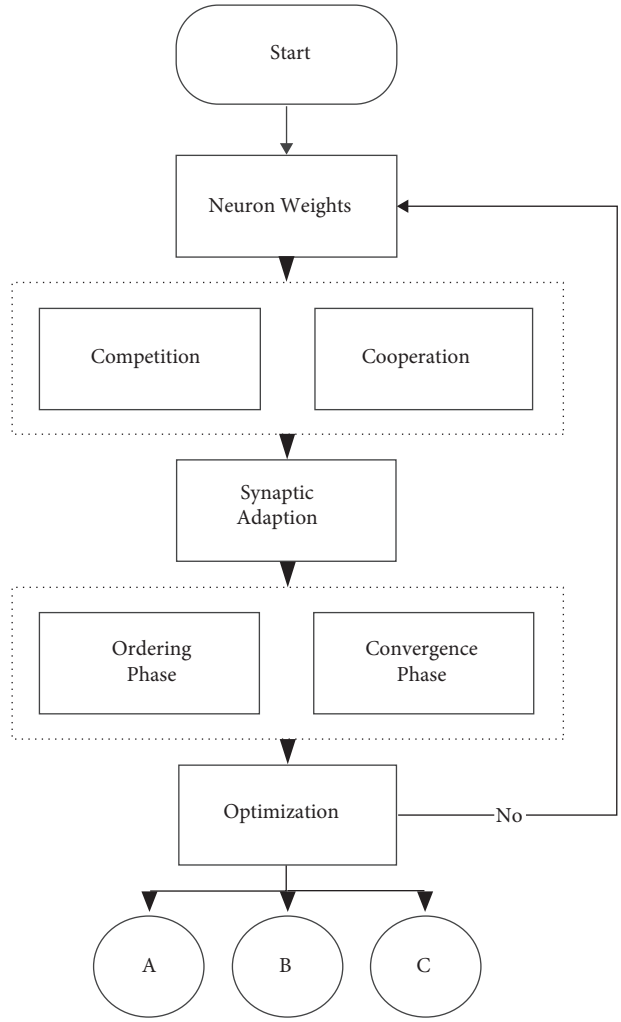


Figure 1: The logical data flow of GsONG.

Table 1: Statistical analysis of data set.

	V1	V2	V3	V4	V5
Mean	21.12	32.08	8.63	2.31	4.17
1st Q	23.09	34.72	9.40	2.35	4.38
Median	24.34	35.19	9.21	2.48	4.63
Mean	24.44	35.87	9.04	2.42	4.60
Std	3.45	4.89	0.97	0.49	0.59
3rd Q	25.51	37.82	9.67	2.73	4.81
Max	26	39	10	3	5

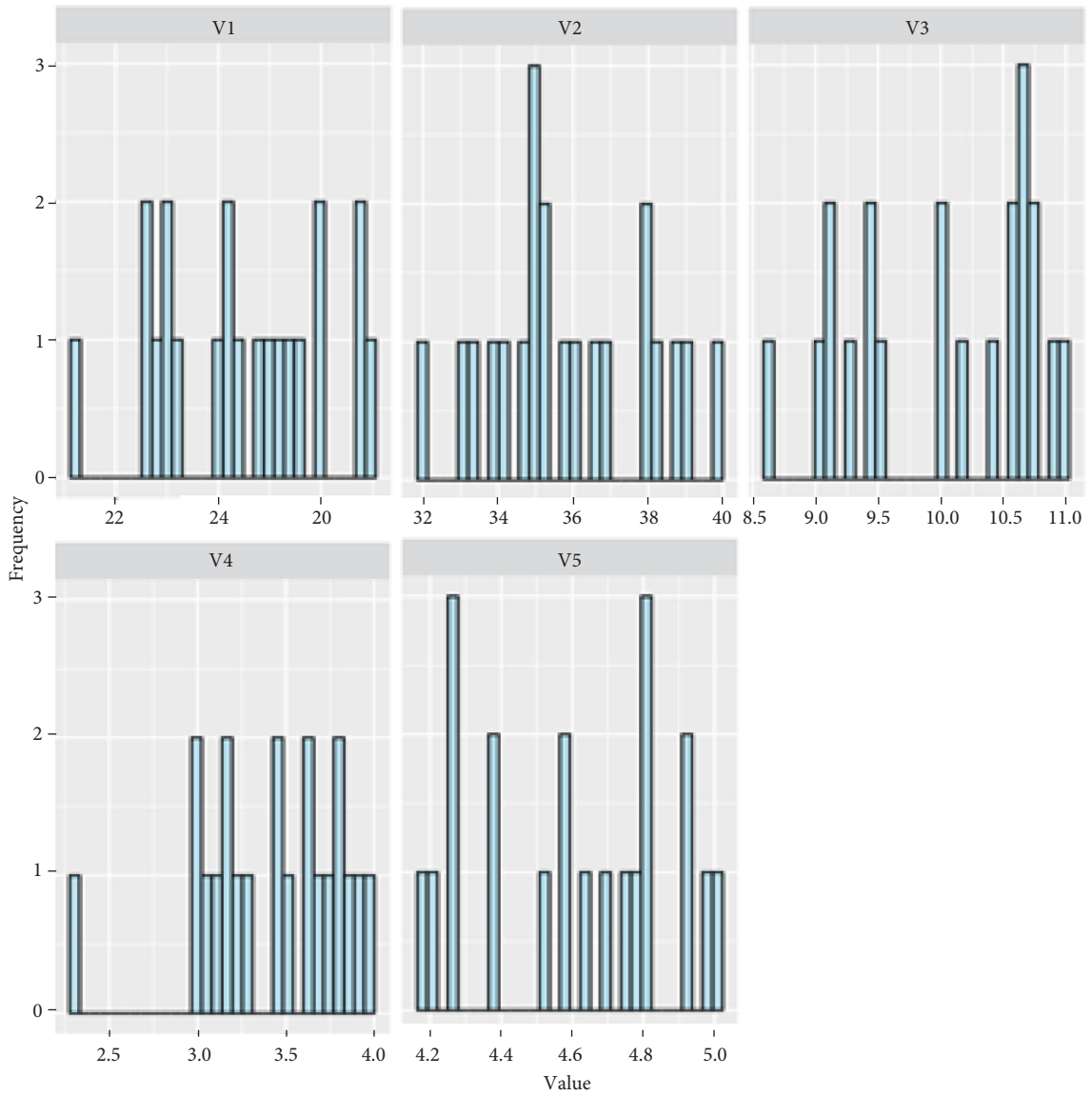


Figure 2: Histogram for each feature of data set.

$$\tau_2 = \frac{n_0}{lnln(100 \cdot \eta_0)} \text{ and } \tau_1 = \frac{n_0}{lnln(\sigma_0)}, \tag{11}$$

where n_0 is the number of repetitions of the phase of the device, η_0 is the initial learning rate, and σ_0 is the initial value of the active amplitude that in turn results from the following relation:

$$\sigma_0 = \sqrt{w^2 + h^2}, \tag{12}$$

where w and h are the length and height of the two-dimensional grid, respectively.

Also, the topological neighborhood function $h_{j,i(n)}$ initially includes almost all competing-level neurons centered on the winning neuron and is gradually limited to a few neurons or even just the winning neuron. Considering a two-dimensional frame, the value of the "radius" of the grid was taken as the initial value σ_0 of the active width and as the value of the parameter τ_1 of the above relation [26]:

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right), n = 0, 1, 2, \ldots \; with \; \tau_1 = \frac{1000}{\log \; \log \sigma_0}. \tag{13}$$

Convergence phase is the phase in which the weights acquire their final values better coordinated in the training examples. In this phase, the number of repetitions was determined by the dimension of the network inputs. The learning rate $\eta(n)$ remained constant at values close to 0.01, and finally, the neighborhood $h_{j,i(n)}$ was limited to the nearest neighbors of the winner-neuron, ending up containing only the winner neuron.

In the second phase (convergence), the values of the learning rate and the active amplitude remained constant and equal to 0.01 and 0.0001, respectively.

Regarding the learning rate, the variables that were selected are λ_i and λ_f that control the rate at which the neural network learns, while ε_i and ε_f define the initial and final rate

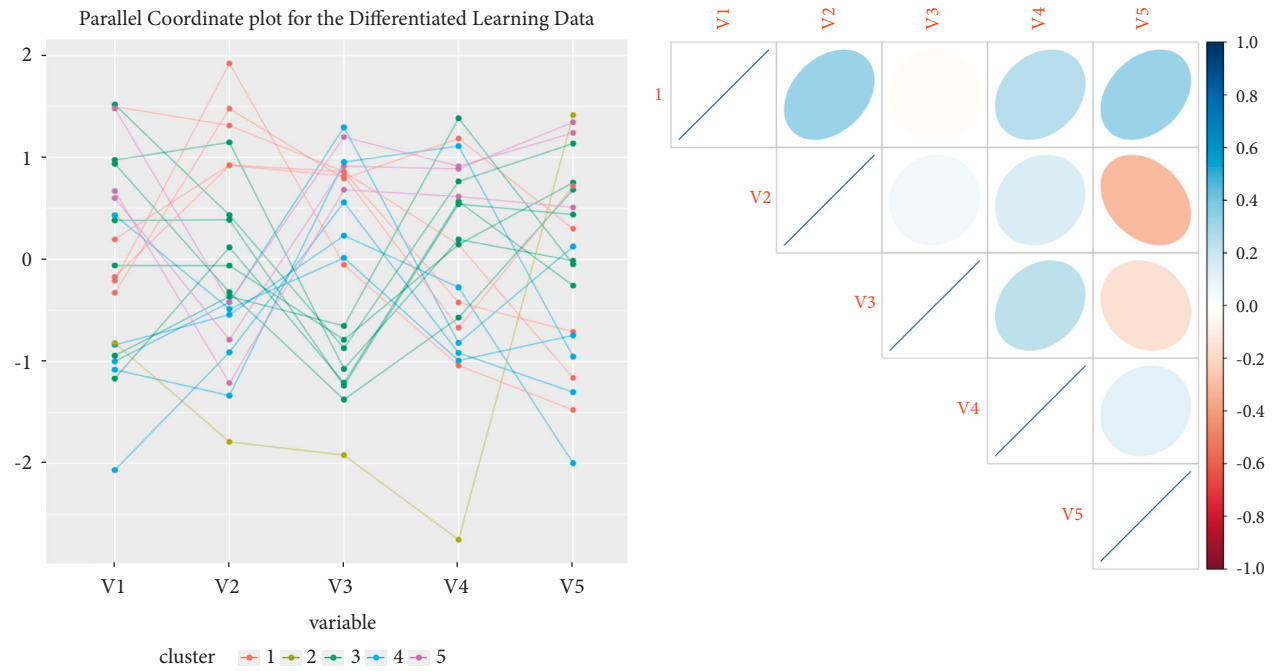


Figure 3: (a) Parallel coordinates plot between features and (b) correlation matrix between features.

according to which the neural network is trained. The variable $t_{\max}$ is the maximum number of execution times. With t the current season, $t_{\max}$ the total number of seasons, $\overrightarrow{x}$ the input signal generated at the beginning of each season, n each network node, n_w the vector that carries each neuron, and k is the degree of each node once, it has been classified into steps. All nodes were sorted in ascending order based on the Euclidean distance of their vectors from the input signal as follows [21, 26]:

$$\left\|\overrightarrow{x} - \overrightarrow{n_w}\right\|^2. \tag{14}$$

The weights of the nodes were adjusted in the order we have arranged them so that [27]

$$\overrightarrow{n_w} \leftarrow \overrightarrow{n_w} + \left[\overrightarrow{n_w} \times e(t) \times h(k) \times \left(\overrightarrow{x} - \overrightarrow{n_w}\right)\right], \tag{15}$$

where

$$h(t) = \exp\left(\frac{-k}{\sigma^2(t)}\right),$$

$$\sigma^2(t) = \lambda_t \times \left(\frac{\lambda_f}{\lambda_i}\right)^{t/t_{\max}}, \tag{16}$$

$$\varepsilon(t) = \varepsilon_t \times \left(\frac{\varepsilon_f}{\varepsilon_i}\right)^{t/t_{\max}}.$$

To optimize the above process, a heuristic method was used that investigates the space of an objective function by regulating the optimal topologies of neurons that form pathway segments in a semi-contemplative manner. Each neuron is attracted to the position of the best location found by the heuristic function and the best location it has encountered, while, at the same time, it tends to move randomly.

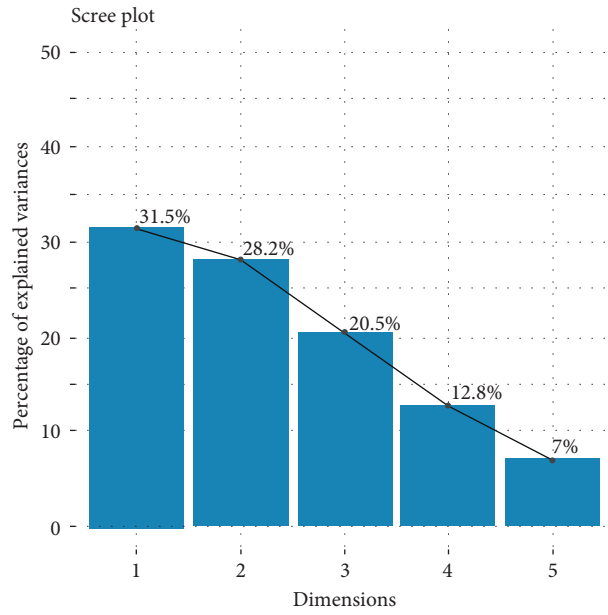


Figure 4: Principle component analysis.

Specifically, when an entity i discovers a locality that is superior to the previous ones it located, then it upgrades it to the best current for i. There is a current best for all n entities at any time t, during iterations. The goal is to find the best overall until the position of the neuron can no longer be improved [28].

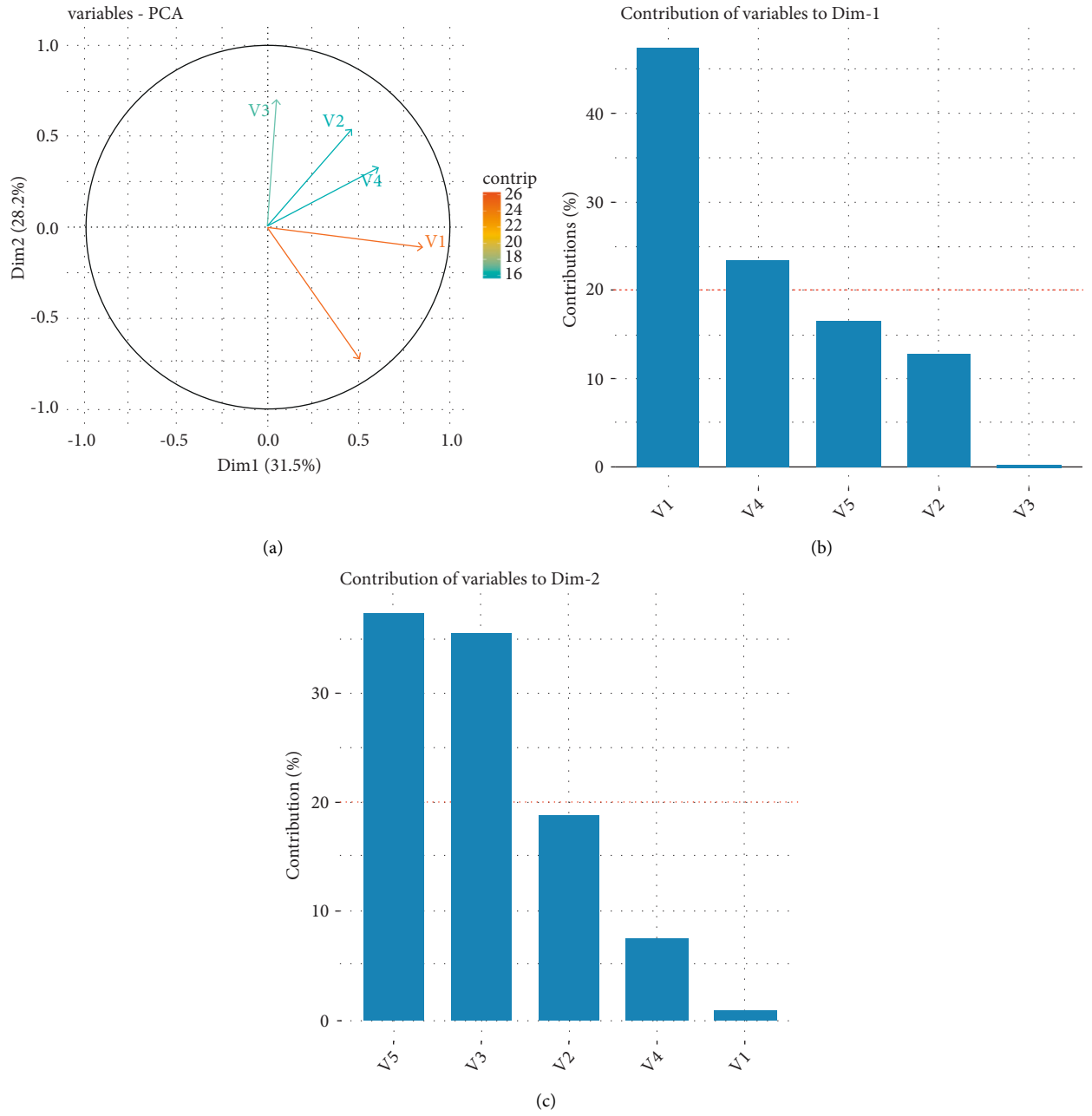


FIGURE 5: (a) Plot of contribution of PCA, (b) plot of contribution Dim-1, and (c) plot of contribution Dim-2.

Let p and u be the position and velocity for entity i, respectively. The new velocity vector is identified by the following formula [29]:

$$u_{n,m}^{\text{new}} = u_{n,m}^{\text{old}} + \Gamma_1 \times r_1 \times \left(p_{n,m}^{\text{local}_{\text{best}}} - p_{n,m}^{\text{old}}\right) + \Gamma_2 \times r_2 \times \left(p_{n,m}^{\text{global}_{\text{best}}} - p_{n,m}^{\text{old}}\right), \quad (17)$$

where $u_{n,m}$ represents the convergence speed, r_1, r_2 represents independent random numbers, Γ_1, Γ_2 represents learning parameters, $p_{n,m}^{\text{local\_best}}$ represents the best local solution, and $p_{n,m}^{\text{global\_best}}$ represents the best total solution.

The heuristic optimization algorithm renews the convergence speed component and then adds speed to the position component. This renewal depends on both the optimal solution/position discovered and the one used by all active neurons. If, at some point, the best solution discovered is better than that of the population, it replaces it. The initial locations of all neurons consist of being evenly distributed so that they are a sample for most areas of the search space. It is also possible for the original vector of an entity to be taken as zero. The new location is described by the following equation [22, 29] (Figure 1):

$$p_{n,m}^{\text{new}} = p_{n,m}^{\text{old}} + u_{n,m}^{\text{new}}, \quad (18)$$

where u is delimited to a range $[0, u_{\max}]$.

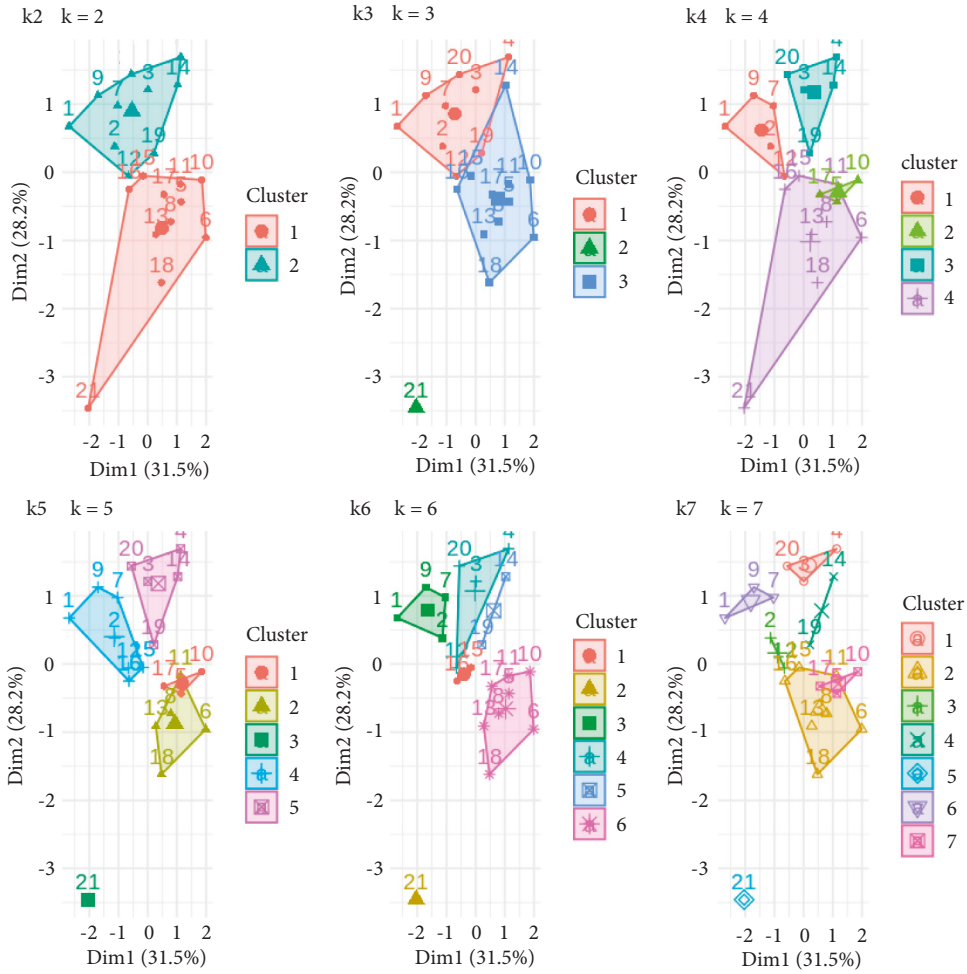


Figure 6: Plot of $k = [2–7]$ centers.

A descriptive illustration of the overall process is presented in the diagram below.

4. Experiments

The aforementioned algorithm was applied to student assessment data to handle the problem of categorizing students into heterogeneous groups with comparable features at the group level, where we assume that we have students who should be classified into groups at most.

The data used relate to quantitative individual performance and psychosocial data of students of a heterogeneous class of students, to implement the multicriteria test procedures.

Specifically, the data refer to the holistic assessment of an elementary school student class with a total capacity of 21 people, where Raven's IQ test was used to assess general mental ability (V1), and the math performance test for primary school students was used to determine mathematical skills (V2). The student's grade point average was used to evaluate performance (V3). The learning disability scale was used to determine social or emotional skills or difficulties (V4). Finally, the psychosocial adjustment tool was used to assess social or emotional skills or difficulties (V5). It is important to emphasize that the problem is trying to be identified on a completely realistic basis, based on the real process that a teacher would follow in a department applying differentiated learning but within the children who are already part of a department. Specifically, the 21 children who participate in this classification process based on the proposed system are a real school class.

Specifically, the 21 children who participate in this classification process based on the proposed system are a real school class. The separation attempted is a realistic approach where the groups of 3, 4, 5, or 6 children that may arise are a fully satisfactory sample of children with homogeneous elements to whom personalized learning techniques can be applied. It should be emphasized that the resulting groups do not form new classes, but groups that receive training materials, instructions, exercises, and so on depending on the level in which they were classified but all within the same classroom.

In order to prove the correct use of this data set, a thorough preprocessing of the data was performed for the purpose of validation checks that prove the reliability of the data set under consideration, before the use of the proposed algorithm. This process is necessary as the initial data often suffer from various kinds of problems, such as conflicting

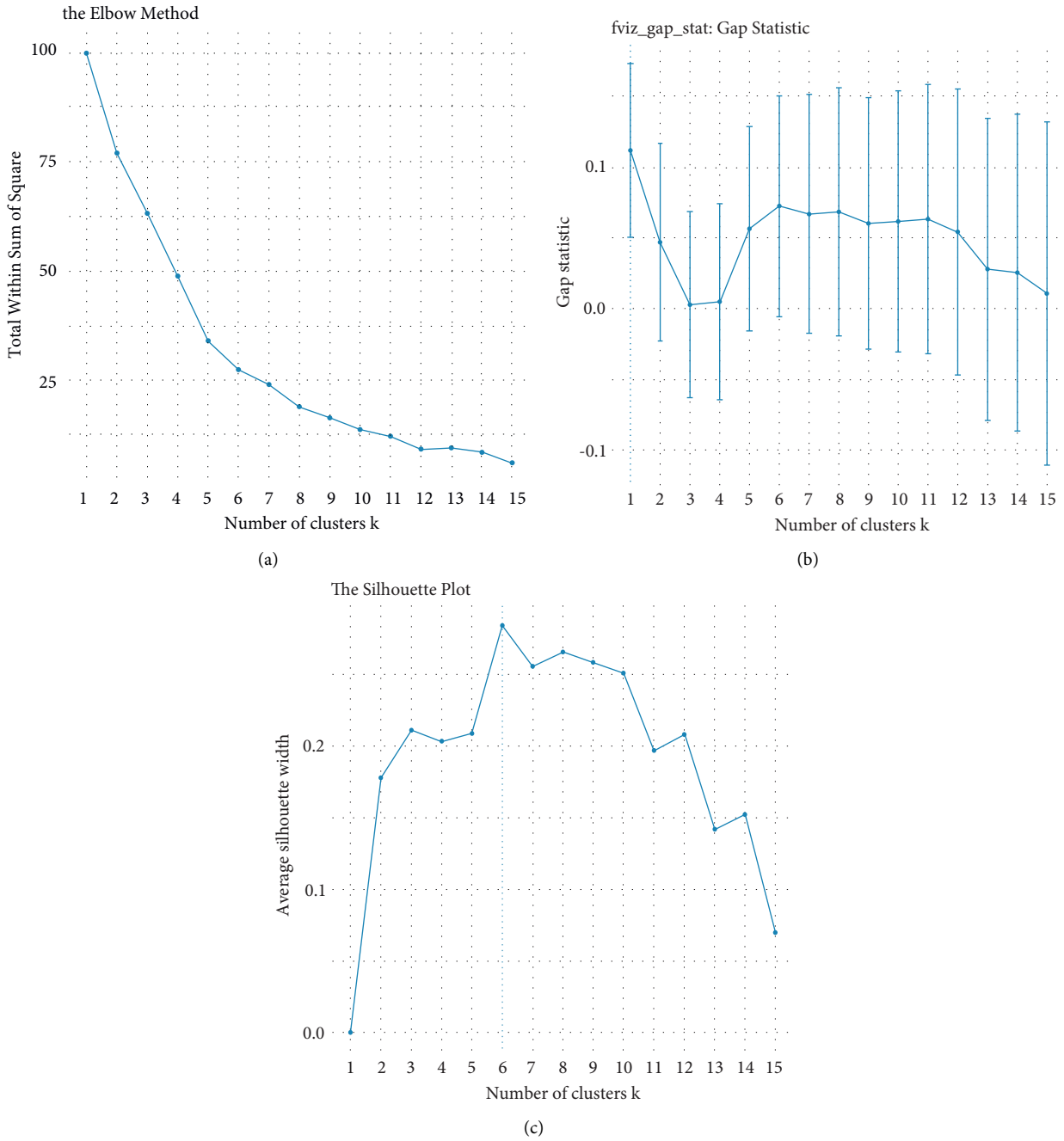


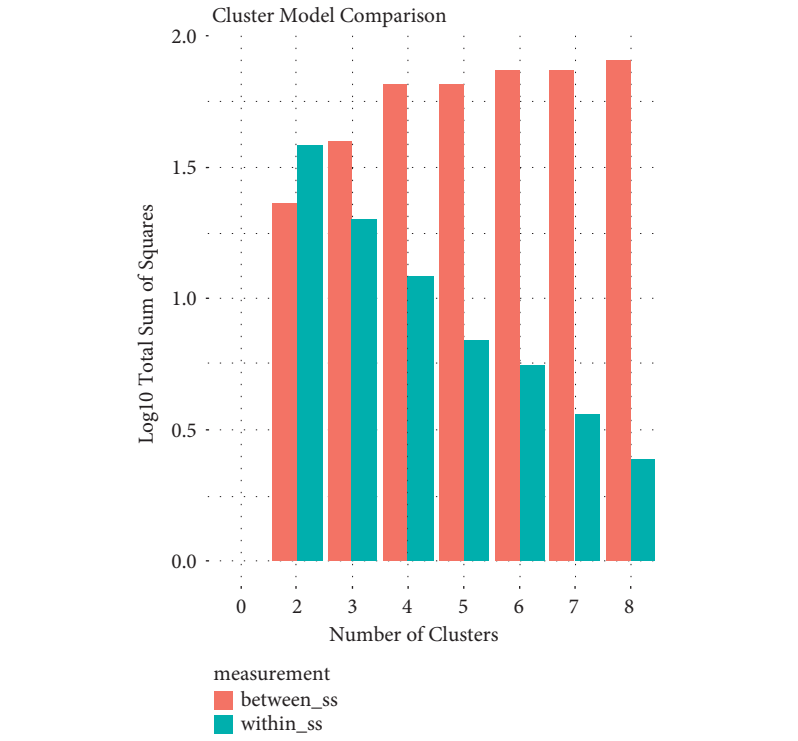
Figure 7: (a) "Elbow" method, (b) "Gap" statistic, and (c) "Silhouette" method.

information, coding inconsistencies, noise, and extremes, but also in addressing specific requirements that require data transformation, such as the discretization, the normalization, the reduction of their dimensions, or the selection of the most appropriate characteristics.

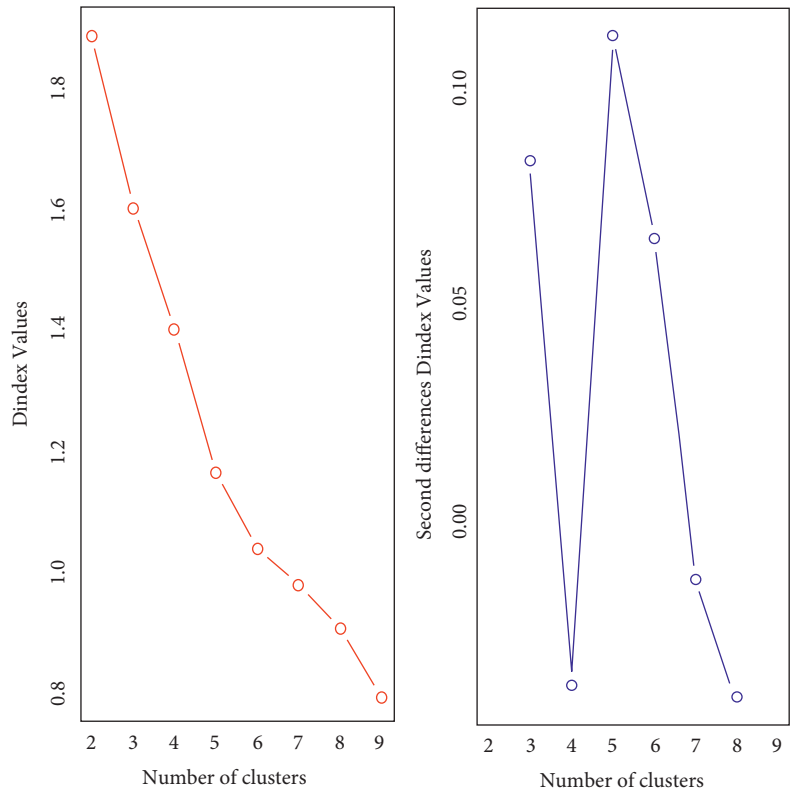
Initially, an indicative statistical analysis of the data set was performed. The main object of the above statistical analysis is the analysis and interpretation of the data used with the ultimate goal of drawing safe conclusions for making correct decisions. Specifically, Table 1 shows the probability for each sample to belong to a specific subset and if the sample space is made up of discrete random variables for which a cumulative probability function can be used to determine the distribution. The statistical analysis of all 21 students is presented in Table 1.

For the clear and distinct localization of the fluctuation of their values, the graphs of the statistical frequencies of the price ranges of each feature used in the data set are presented in histograms. The height of each region is equal to the ratio of the frequency to the range of values represented by the rectangle. All five features are presented in Figure 2.

Because the data set used is multivariate numeric data, the parallel coordinates plot is listed in Figure 3(a), which is an imaging technique that facilitates the comparison of



(a)



(b)

Figure 8: Continued.

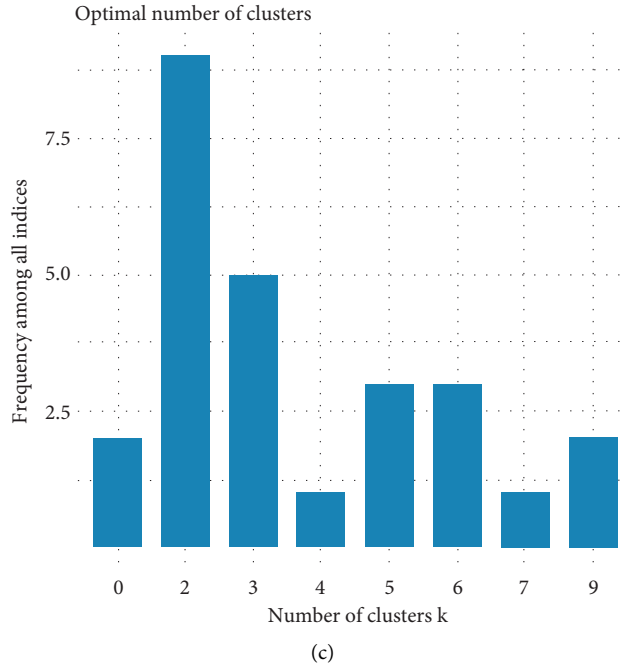


Figure 8: (a) Sum of squares method, (b) Clustree method, and (c) NbClust method.

multiple quantitative variables simultaneously in order to identify patterns, similarities, complexes, and positive and negatives or neutral data relationships.

To investigate the two-way relationships between the features of the data set, a correlation analysis was performed, and Figure 3(b) shows the resulting correlation matrix.

Correlations are useful because they can indicate a predictive relationship that can be exploited in practice, although statistical dependence is not sufficient to prove the presence of a causal relationship (i.e., the correlation does not imply causality). A principle component analysis (PCA) test was then performed to detect data covariance and to apply if parameter reduction is required. As can be seen from the scree plot in Figure 4, the principal components retain less than 60% of the statistical data from the original data, so no parameter reduction is required.

From Figures 5(a)–5(c), it is understood that V1 and V2 tend to increase together in the first dimension, while in the second dimension, V1, V2, and V3 increase together. These two groups of features have a homogeneous and corresponding correlation.

The process of pretreatment of the set performed proves and ensures the quality of the data to be used by the proposed algorithm. Then, in order to identify the appropriate groups of students that will be the uniform clusters of differentiated learning, sequential analyzes were performed with various clustering methods (such as k-means, k-Medoids, and k-Centroids) [17, 22]. Specifically, successive configurations were performed with the available data, from 2 to 7 groups (clusters centers), in order to identify the best. For example, in the example of Figure 6 where he presents the configurations using the k-means algorithm, as this algorithm is sensitive to the initial positions of the

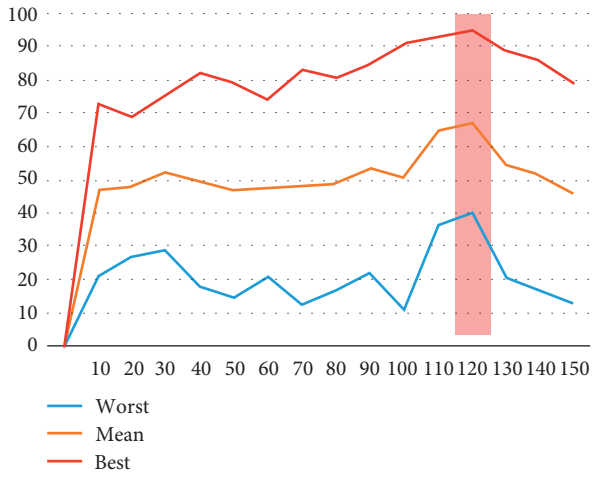


Figure 9: Optimization parameters.

centers of the clusters, 10 initial configurations were created, and then, all the results were calculated on average.

The above visual assessment gives clear explanations of where the demarcations between clusters occur; however, no information is given on the optimal number of clusters. To determine the optimal number of clusters, the method "Elbow" was used sequentially, in which the sum of the squares for each number of blocks is calculated and formed and the optimal number results in the abrupt change of inclination (Elbow), as in Figure 7(a).

"Gap" statistics method is also used that compares the total variance within clusters for different center values with their expected values under zero data reference distribution. The estimation of the optimal clusters is the value that

maximizes the statistical element of the gap, that is, that gives the largest statistical gap, which means that the clustering structure is far from the random uniform distribution of points. The "Gap" statistic is shown in Figure 7(b).

The "Silhouette" method was also used that calculates the average silhouette of the observations for different clusters values. The optimal number of blocks is the one that maximizes the average silhouette in a range of possible values. The "Silhouette" method is shown in Figure 6(c).

Another validation method used is to select the optimal number of clusters by minimizing the sum of squares within a cluster (how tight each cluster is) and by maximizing the sum of squares between the clusters (how sparsely the clusters are distributed). This method is shown in Figure 8(a).

Also, the Clustree statistical method produces a single score that takes into account only one set of clusters at a time considering how the samples change groups as the number of clusters increases. This is useful for showing which groups are different and which are unstable. The methodology is shown in Figure 8(b).

Another very interesting measurement comes from the NbClust method for determining the relative number of clusters, which proposes the best scheme from the different results obtained from the evaluation of 30 indicators.

The specific measurement is shown in Figure 8(c).

A Davies–Bouldin index was used to evaluate the candidate solutions [17, 22]. We consider that $R_{i,j}$ is an evaluation measure of each cluster that is calculated by the following equation:

$$R_{i,j} = \frac{s_i + s_j}{M_{i,j}}, \tag{19}$$

where s_i, s_j are the dispersions of the i and j blocks, respectively, which are calculated from the following equation:

$$s_i = \left(\frac{1}{T_i} \sum_{j=1}^{T_i} \left| X_j - A_i \right|^q \right)^{1/q}, \tag{20}$$

where T_i is the number of vectors in the i block, X_j is the vector of each student's attributes, and A_i is the center of the i block. Each student is classified in the cluster whose center is closer to his own. $M_{i,j}$ is the Minkowski metric for the distance of i and j blocks, which are calculated from the following equation [21, 30, 31]:

$$M_{i,j} = \left\{ \sum_{k=1}^{N} \left| a_{k,i} - a_{k,j} \right|^p \right\}^{1/p}, \tag{21}$$

where $a_{k,i}$ is the k element of A_i and $a_{k,j}$ is the k element of A_j. The value of the objective function is finally defined as follows:

$$\underline{R} = \frac{1}{N} \sum_{i=1}^{N} R_i, \tag{22}$$

where R_i is the maximum value of $R_{i,j}$ for $i \neq j$.

If a group of students was too small or too large, an error was introduced into the objective function, doubling its value, making all groups four to six students.

Table 2: Optimization process results.

Parameters	Worst	Mean	Best
10	21	47	73
20	27	48	69
30	29	53	76
40	18	50	82
50	15	47	79
60	21	48	74
70	13	48	83
80	17	49	81
90	22	54	85
100	11	51	91
110	36	65	93
120	40	68	95
130	21	55	89
140	17	52	86
150	13	46	79

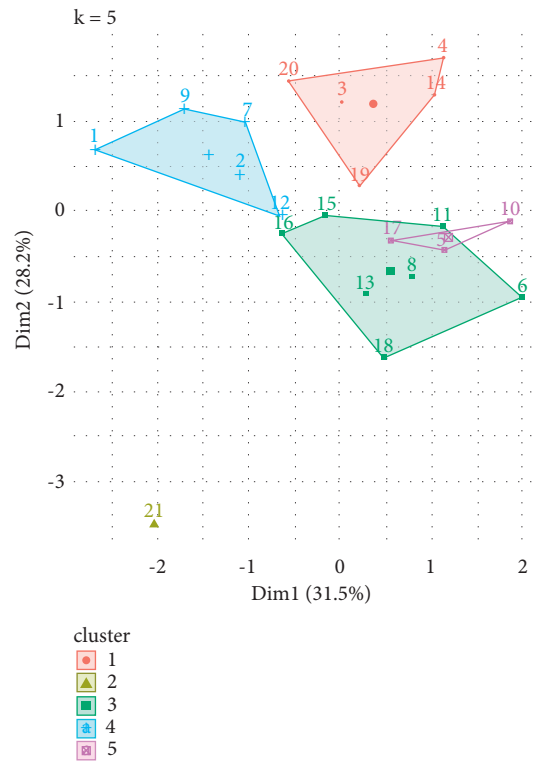


Figure 10: Clusters distribution.

The number of participants in a collaborative student group should be between four and six, as this allows good cooperation and communication among the members.

As a result, k was calculated as an integer consistently greater than or equal to the quotient of dividing the number of students by the number four. Following the completion of the group separation using the suggested algorithm, the solutions were assessed intragroup for homogeneity using the coefficient of variation (CV).

The CV is an index of relative variance or dispersion, which expresses the homogeneity of a set of measurements of values of a random quantitative variable and the accuracy of an experimental design.

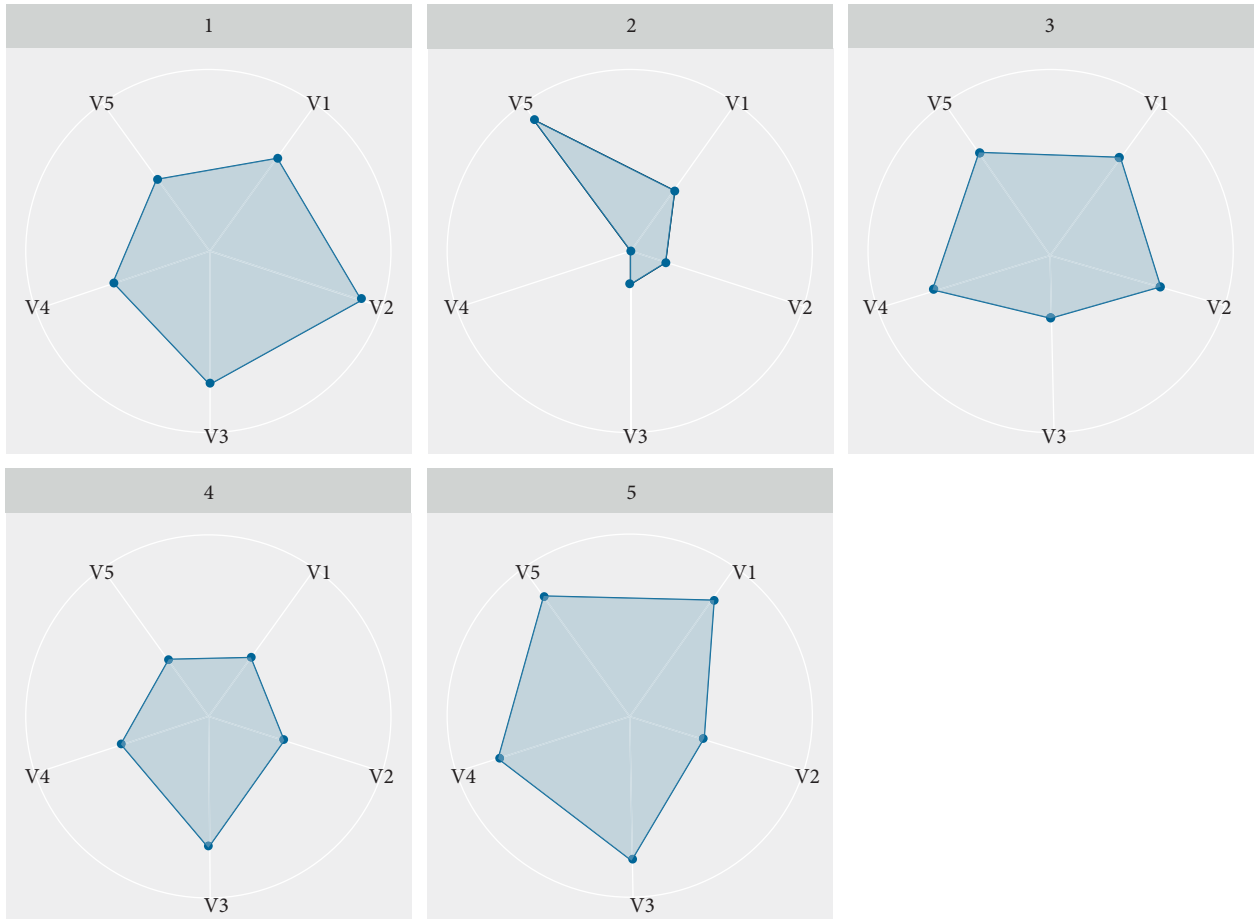


Figure 11: Differentiated learning attributes.

The following ratio was used to calculate the index for sample data:

$$CV = \frac{S}{\underline{Y}\,100,} \tag{23}$$

where S is the standard deviation and Y is the arithmetic mean of the sample measurements.

Values close to zero indicate homogeneity in terms of characteristics, while values close to 1 indicate inhomogeneity. In general, the values of the CV index show the level of homogeneity as follows:

(1) High $(0.00 < \mathrm{CV} \leq 0.25)$

(2) Medium $(0.25 < \mathrm{CV} \leq 0.40)$

(3) Low $(\mathrm{CV} > 0.40)$

The Kruskal–Wallis test [32] was used to determine if the resulting groups differed at different levels of statistical significance (0.01, 0.001).

The nonparametric Kruskal–Wallis test was used for the nonparametric analysis of variance in independent samples and was selected as in this case, the condition of normality of the populations examined is not met, the sample is small (<20 for each cluster), and the values of the dependent variables do not express quantity but are ranks where individuals are simply ordered according to some criterion.

The way to calculate the acceptance or rejection of the null hypothesis that the random samples are homogeneous is to examine whether the quantities R_i/n_i, $i = 1, \ldots, k$, $k > 3$ are approximately equal to each other and equal to $(n + 1)/2$ or if the following ratio is close to zero:

$$\sum_{i=1}^{k} \left(\frac{R_i}{n_i} - \frac{(n+1)}{2} \right)^2. \tag{24}$$

The magnitude of the size effects was calculated using the Eta Squared η^2 and Cohen's d indicators [33]. Implementing the clustering process, initially, the parameters of the proposed algorithm were randomly initialized, based on the description performed above.

To find the optimal values for which the algorithm performs best, extensive trial and error tests were performed for different hyperparameters of the optimization algorithm. Initially, the population of optimal solutions was tested by testing values from 10 to 150 with a progressive increase of 10 units.

The algorithm performed 10 iterations for each value, whose diagrammatic representation of the results is presented in Figure 9, where the best, worst, and average values of their results are stated.

As it is understood, the optimal parameters that the algorithm shows greater convergence are for 120 particles,

Table 3: Mean and standard deviation of clusters.

Cluster		0 (6 studs)	1 (5 studs)	2 (4 studs)	3 (6 studs)	4 (6 studs)
V1	Mean	18.86	15.711	25.23	21.47	21.39
	StDev	0.77	1.18	0.94	1.08	1.07
V2	Mean	30.74	27.69	37.81	32.94	32.96
	StDev	1.02	0.44	0.98	1.35	1.34
V3	Mean	8.42	6.78	9.99	9.43	9.51
	StDev	0.53	0.47	0.03	0.56	0.57
V4	Mean	2.01	1.03	2.98	3.02	3.04
	StDev	0.02	0.46	0.01	0.01	0.01
V5	Mean	3.84	2.79	5.05	4.96	4.89
	StDev	0.77	0.48	0.02	0.04	0.05

Table 4: Coefficient of variation of each cluster.

Cluster	V1	V2	V3	V4	V5
0 (6 students)	0.04	0.03	0.06	0.01	0.02
1 (5 students)	0.06	0.02	0.05	0.03	0.01
2 (4 students)	0.04	0.03	0.01	0.01	0.01
3 (6 students)	0.05	0.04	0.05	0.01	0.02
4 (6 students)	0.06	0.05	0.05	0.01	0.02

which were selected for the further clustering process. Also, Table 2 presents in detail the above values.

The resulting clusters are shown in Figure 10.

Also, Figure 11 shows the exact distribution of the differentiated learning attributes.

The clusters created by the proposed algorithm with the average values per rating scale are presented in Table 3.

The values of the CVs, for the evaluation of the homogeneity within each cluster formed through the proposed algorithm, yielded values from 0.00 to 0.06, which are presented in Table 4.

Finally, the results of the Kruskal–Wallis test showed that the five groups differ significantly in [21]:

(1) The general mental capacity (V1), $[\chi^2(3) = 18,67, p < 0,001, d = 6,9, \eta^2 = 0,92]$

(2) Mathematical performance detection test (V2), $[\chi^2(3) = 18,23, p < 0,001, d = 5,9, \eta^2 = 0,90]$

(3) The quarter points that are an estimate of their performance (V3), $[\chi^2(3) = 17,62, p < 0,01, d = 4,9, \eta 2 = 0,86]$

(4) The scale of detection of learning difficulties (V4), $[\chi^2(3) = 19,26, p < 0,001, d = 9,4, \eta^2 = 0,96]$

(5) Their psychosocial adaptation characteristics (V5), $[\chi^2(3) = 16,92, p < 0,01, d = 4,3, \eta^2 = 0,82]$

5. Discussion and Conclusions

Based on the results obtained and presented in detail above, it is obvious that the utilization of the proposed algorithm can find a reliable solution to the extremely difficult problem of creating and forming student groups for the implementation of individualized teaching programs. The methodology proved that through the widespread use of intelligent methods, small and heterogeneous groups of students can emerge with the members of each group sharing similar features in terms of student ability [34], learning challenges [35], and psychosocial and cognitive profile [36]. In this way, in addition to being able to quickly manage the student potential in their class and knowing the individual characteristics of each group, the teacher can easily manage the student potential of their class [37]; he can offer high-quality education, through differentiated approaches that take into account the special educational needs and capabilities of each group, their particular interests, their unique experiences, their learning rhythms, their learning style, their cultural background, and their self-perception [38]. Also, as a clustering approach, the algorithm can be used in both traditional classrooms and digital or e-learning programs, facilitating the educational role [39], as it can compensate for difficulties in multicriteria grouping and differentiation of students in a variety of subjects [40]. Another significant benefit of the method is that it may be used with a large number of students and deliver results in a short period of time, provided of course there is the appropriate data for processing [41]. Another supporting presumption is that there is no limit to the data that can be accepted as quantitative data or to the evaluable factors that result from the multifaceted and holistic assessment of the student [42].

In this study, one limitation was the small number of participants, which may raise validity issues. However, the algorithmic approach used was weighted to compensate for any psychometric issues, and it is important to emphasize—which gives the method applicability in real conditions—that the application of computational methods was done in real order contexts, where there are physical limitations to a maximum number of students attending them.

From a technical point of view, the algorithm presented, which is proposed for the first time in the literature, shows a very high degree of convergence, which is evidenced by the very high clustering results that were achieved and confirmed experimentally. A very important observation also concerns the fact that the optimization method used converges very quickly, while, in all the tests, it was not observed to be trapped in local optimal, thereby avoiding incorrect cluster formations. The principle of differentiated learning is a modern educational method that aims to offer high-quality education, through differentiated approaches that take into account the special educational needs and capabilities of each student, their special interests, their unique experiences, their learning rhythms, their learning style, their cultural background, and their self-perception. Even, in this case, however, the level of students is never the same, resulting in the adaptation of teaching to the different levels of learning ability that exist within a classroom. The internal differentiation required in these cases should include a wide variety of practices and individualized forms of organizing the learning process. In that vein, this paper presented an innovative and fully efficient differentiated learning framework. It is an intelligent system that can classify students into similar homogeneous groups, based on their general mental ability, the performance of their student skills, grade points, learning difficulties they may face, and

finally the criteria of psychosocial adjustment for the assessment of skills and their school adaptation to the school environment. It is based on advanced engineering learning techniques for performing high-level analyzes for the effective reorganization of educational learning systems based on evaluation criteria. The implementation of the proposed algorithm is based on the ideal use and combination, for the first time in the machine learning literature, of the two well-known clustering methodologies (cooperation and competition) in order to produce an extremely efficient and fast neural system. The proposals for the continuation of this research focus mainly on the investigation and extension of the model with inherent capabilities of natural language processing, for the automated system to fully utilize the capabilities of the wider dependencies of modeling learning systems, with greater accuracy and efficiency. The future study of the effect of such a grouping methodology on the student's learning development in comparison to traditional methods of separation is also intriguing, as is the realization of such research using nonparametric machine learning methods.

Data Availability

The data used in the paper are available upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] H. J. Cha and M. L. Ahn, "Development of design guidelines for tools to promote differentiated instruction in classroom teaching," *Asia Pacific Education Review*, vol. 15, no. 4, pp. 511–523, 2014.

[2] X. Zhu, C. M. Cheong, G. Y. Li, and J. Wu, "Primary school teachers' conceptions of reading comprehension processes and its formulation," *Frontiers in Psychology*, vol. 11, p. 615, 2020.

[3] V. Demertzi and K. Demertzis, *A Hybrid Adaptive Educational eLearning Project Based on Ontologies Matching and Recommendation System*, http://arxiv.org/abs/2007.14771, 2021.

[4] A. Demetriou, G. Spanoudis, and A. Mouyi, "Educating the developing mind: towards an overarching paradigm," *Educational Psychology Review*, vol. 23, no. 4, pp. 601–663, 2011.

[5] M. Nedashkovskyy, G. Śmigielski, O. Sokolov, and A. Mreła, "Visual methods of processing survey data in social disciplines based on fuzzy logic," *Bulletin of the Polish Academy of Sciences*, vol. 69, no. 5, Article ID e138812, 2021, https://journals.pan.pl/dlibra/publication/138812/edition/120730.

[6] F. Essalmi, L. Jemni Ben Ayed, M. Jemni, S. Graf, and Kinshuk, "Generalized metrics for the analysis of E-learning personalization strategies," *Computers in Human Behavior*, vol. 48, pp. 310–322, 2015.

[7] A. Klašnja-Milićević and M. Ivanović, "E-learning personalization systems and sustainable education," *Sustainability*, vol. 13, no. 12, p. 6713, 2021.

[8] S. Monemian, *A Neuroevolutionary Neural Network-Based Collaborative Filtering Recommendation System," Thesis*Laurentian University of Sudbury, Sawston, UK, 2020, https://zone.biblio.laurentian.ca/jspui/handle/10219/3503.

[9] I. Rodríguez-Ardura and A. Meseguer-Artola, "What leads people to keep on e-learning? An empirical analysis of users' experiences and their effects on continuance intention," *Interactive Learning Environments*, vol. 24, no. 6, pp. 1030–1053, Aug. 2016.

[10] I. Rodríguez-Ardura and A. Meseguer-Artola, "Flow experiences in personalised e-learning environments and the role of gender and academic performance," *Interactive Learning Environments*, vol. 29, no. 1, pp. 59–82, Jan. 2021.

[11] J. T. Nganji, "Towards learner-constructed e-learning environments for effective personal learning experiences," *Behaviour & Information Technology*, vol. 37, no. 7, pp. 647–657, Jul. 2018.

[12] S. Maghsudi, A. Lan, J. Xu, and M. van der Schaar, "Personalized education in the artificial intelligence era: what to expect next," *IEEE Signal Processing Magazine*, vol. 38, no. 3, pp. 37–50, May 2021.

[13] K. Demertzis, D. Tsiotas, and L. Magafas, "Modeling and forecasting the COVID-19 temporal spread in Greece: an exploratory approach based on complex network defined splines," *International Journal of Environmental Research and Public Health*, vol. 17, no. 13, p. 4693, Jan. 2020 Art. no. 13.

[14] Y. Contoyiannis, S. G. Stavrinides, M. P. Hanias, M. Kampitakis, P. Papadopoulos, and S. Potirakis, "On the effectiveness of imposing restrictive measures in a graded Self-Organized Criticality epidemic spread model the case of COVID-19," May 2020, http://arxiv.org/abs/2004.00682.

[15] K. Demertzis, D. Taketzis, D. Tsiotas, L. Magafas, L. Iliadis, and P. Kikiras, "Pandemic analytics by advanced machine learning for improved decision making of COVID-19 crisis," *Processes*, vol. 9, no. 8, p. 1267, Aug. 2021.

[16] S. Guzdek, "The importance of clusters for the region's economy," *Biul. KPZK*, vol. 2018, no. 272, pp. 54–64, 2018, https://journals.pan.pl/dlibra/publication/128619/edition/112206.

[17] S. Theodoridis and K. Koutroumbas, "Clustering algorithms III: schemes based on function optimization," in *Pattern Recognition*, S. Theodoridis and K. Koutroumbas, Eds., pp. 701–763, Academic Press, Boston, Fourth Edition, 2009.

[18] K. M. R. Alam, N. Siddique, and H. Adeli, "A dynamic ensemble learning algorithm for neural networks," *Neural Computing & Applications*, vol. 32, no. 12, pp. 8675–8690, 2020.

[19] L. Starkel, "Clustering of extreme rainfalls and evolution of fluvial systems in the Holocene," *Studia Quaternaria*, vol. 23, pp. 23–28, 2006, https://journals.pan.pl/dlibra/publication/136653/edition/119393.

[20] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, Dec. 2006.

[21] I. Kononenko and M. Kukar, "Chapter 12 - Cluster Analysis," in *Machine Learning and Data Mining*, I. Kononenko and M. Kukar, Eds., Woodhead Publishing, Sawston, United Kingdom, pp. 321–358, 2007.

[22] G. Peters, M. Lampart, and R. Weber, "Evolutionary rough k-medoid clustering," in *Transactions on Rough Sets VIII*, J. F. Peters and A. Skowron, Eds., Springer, Berlin, Heidelberg, pp. 289–306, 2008.

[23] M. Ragni and S. Neubert, "Analyzing Raven's intelligence test: cognitive model, demand, and complexity," in *Computational Approaches to Analogical Reasoning: Current Trends*, H. Prade

and G. Richard, Eds., Springer, Berlin, Heidelberg, pp. 351–370, 2014.

[24] P. A. Wilson and B. Lewandowska-Tomaszczyk, "Cognitive structure and conceptual clusters of emotion terms," *Filoz. Nauka*, vol. 2019, pp. 91–123, 2019, https://journals.pan.pl/dlibra/publication/131193/edition/114592.

[25] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin Heidelberg, 3rd ed edition, 2001.

[26] *A New Growing Neural Gas for Clustering Data Streams | Neural Networks*, vol. 78, 2021, https://dl.acm.org/doi/abs/10.1016/j.neunet.2016.02.003.

[27] W. Drozd, "Cluster analysis in research of accident rate in construction sector," *Archives of Civil Engineering*, vol. 64, no. 3, pp. 159–172, 2018, https://journals.pan.pl/dlibra/publication/132319/edition/115603.

[28] P. Golińska, "Information management supporting multimodal transport utilization in virtual clusters," *Management and Production Engineering Review*, vol. 2013, no. 1, 2013, https://journals.pan.pl/dlibra/publication/103319/edition/89327.

[29] J. C. Bansal, H. Sharma, K. Deep, K. N. Das, and A. Nagar, "Special issue on swarm intelligence and its applications to engineering," *International Journal of System Assurance Engineering and Management*, vol. 9, no. 4, pp. 739-740, Aug. 2018.

[30] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent binary classification with generalized performance metrics," *Advances in Neural Information Processing Systems*, vol. 27, 2014, https://papers.nips.cc/paper/2014/hash/30c8e1ca872524fbf7ea5c519ca397ee-Abstract.html.

[31] M. Bhaskar and C. Nick, *An Introduction to Neural Information Retrieval*, https://ieeexplore.ieee.org/document/8620670/, 2018.

[32] J. C. W. Rayner and G. Livingston, "The Kruskal-Wallis tests are Cochran-Mantel-Haenszel mean score tests," *Metron*, vol. 78, no. 3, pp. 353–360, Dec. 2020.

[33] M. A. Adams and T. L. Conway, "Eta squared," in *Encyclopedia of Quality of Life and Well-Being Research*, A. C. Michalos, Ed., Springer Netherlands, Dordrecht, pp. 1965-1966, 2014.

[34] J. S. Tahiri, S. Bennani, and M. K. Idrissi, "An assessment system adapted to differentiated learning within Massive Open Online Courses using psychometric testing," in *Proceedings of the 2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pp. 1–7, Istanbul, Turkey, 8-10, Sep. 2016.

[35] P. Wu and L. Sun, "Research on the cultivation of autonomous learning ability based on SPOC differentiated teaching mode: take "engineering contract management" as an example," in *Proceedings of the 2020 International Conference on Wireless Communications and Smart Grid (ICWCSG)*, pp. 408–411, Qingdao, China, 12-14, Jun. 2020.

[36] A. M. F. Yousef and A. R. Khatiry, "Cognitive versus behavioral learning analytics dashboards for supporting learner's awareness, reflection, and learning process," *Interactive Learning Environments*, vol. 0, no. 0, pp. 1–17, Dec. 2021.

[37] V. S. Varanasi and S. Chilukuri, "Adaptive differentiated edge caching with machine learning for V2X communication," in *Proceedings of the 2019 11th International Conference on Communication Systems Networks (COMSNETS)*, pp. 481–484, Bengaluru, India, 7-11 Jan. 2019.

[38] A. S. Zoolfakar and F. Sulaiman, "Providing differentiated learning experiences through e-Learning on Solid State Devices course," in *Proceedings of the 2009 International Conference on Engineering Education (ICEED)*, pp. 138–142, Kuala Lumpur, Malaysia, 7-8 Sep. 2009.

[39] A. Qodad, Y. Z. Seghroucheni, M. Al Achhab, M. El Yadari, A. El Kenz, and A. Benyoussef, "An adaptive learning system based on a Job model, the differentiated instruction and Felder and Silverman's learning styles model," in *Proceedings of the 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pp. 506–510, Tangier, Morocco, 24-26 Jul. 2016.

[40] N. Aljowaysir, T. O. Ozdemir, and T. Kim, "Differentiated learning patterns with mixed reality," in *Proceedings of the 2019 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 1–4, New Haven, CT, USA, 18-21, Jun. 2019.

[41] M. Andergassen, G. Ernst, V. Guerra et al., "The evolution of e-learning platforms from content to activity based learning: the case of Learn@Wu," in *Proceedings of the 2015 International Conference on Interactive Collaborative Learning (ICL)*, pp. 779–784, Firenze, Italy, 20-24, Sep. 2015.

[42] T. Keuning and M. van Geel, "Differentiated teaching with adaptive learning systems and teacher dashboards: the teacher still matters most," *IEEE Transactions on Learning Technologies*, vol. 14, no. 2, pp. 201–210, Apr. 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 5303872, 11 pages
https://doi.org/10.1155/2022/5303872

Research Article

Extracting Wetland Type Information with a Deep Convolutional Neural Network

XianMing Guan,[1,2,3] **Di Wang**,[4] **Luhe Wan**,[1,2] **and Jiyi Zhang**[5,6]

[1]*Heilongjiang Province Key Laboratory of Geographical Environment Monitoring and Spatial Information Service in Cold Regions, Harbin Normal University, Harbin 150025, China*
[2]*College of Geographic Science, Harbin Normal University, Harbin 150025, China*
[3]*Heilongjiang Geomatics Center of NASMG, Harbin 150081, China*
[4]*Heilongjiang Institute of Geomatics Engineering, Harbin 150081, China*
[5]*College of Geographic Science, Nantong University, Nantong 226019, China*
[6]*Department of Geographic Information Science, Chuzhou University, Chuzhou 239000, China*

Correspondence should be addressed to Luhe Wan; wanluhe@hrbnu.edu.cn

Received 17 December 2021; Revised 1 April 2022; Accepted 19 April 2022; Published 18 May 2022

Academic Editor: Ripon Chakrabortty

Wetlands have important ecological value. The application of wetland remote sensing is essential for the timely and accurate analysis of the current situation in wetlands and dynamic changes in wetland resources, but high-resolution remote sensing images display nonobvious boundaries between wetland types. However, high classification accuracy and time efficiency cannot be guaranteed simultaneously. Extraction of wetland type information based on high-spatial-resolution remote sensing images is a bottleneck that has hindered wetland development research and change detection. This paper proposes an automatic and efficient method for extracting wetland type information. First, the object-oriented multiscale segmentation method is used to realize the fine segmentation of high-resolution remote sensing images, and then the deep convolutional neural network model AlexNet is used to classify automatically the types of wetland images. The method is verified in a case study involving field-measured data, and the classification results are compared with those of traditional classification methods. The results show that the proposed method can more accurately and efficiently extract different wetland types in high-resolution remote sensing images than the traditional classification methods. The proposed method will be helpful in the extension and application of wetland remote sensing technology and will provide technical support for the protection, development, and utilization of wetland resources.

1. Introduction

With the development of remote sensing technology, the quality of available remote sensing images is increasing, and information on ground objects is becoming increasingly detailed; however, the classification accuracy of the traditional pixel-based classification method is low due to the large amount of data, and it can no longer meet the classification requirements of high-resolution remote sensing images [1]. Shallow structure models, such as neural networks and support vector machine ensembles, combine spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery [2]. However, due to the high resolution of remote sensing images, the heterospectrum phenomenon is very obvious. For complex sample structures, shallow structure models are affected by computing power and are unable to learn the sample information adequately [3]. A deep learning network is a deep structure model composed of multiple nonlinear mappings, has a strong function expression ability, can learn more complex training samples, and has good robustness for the classification of complex features in remote sensing images [4, 5].

Wetlands play an important role in regulating the natural environment; they are considered a unique

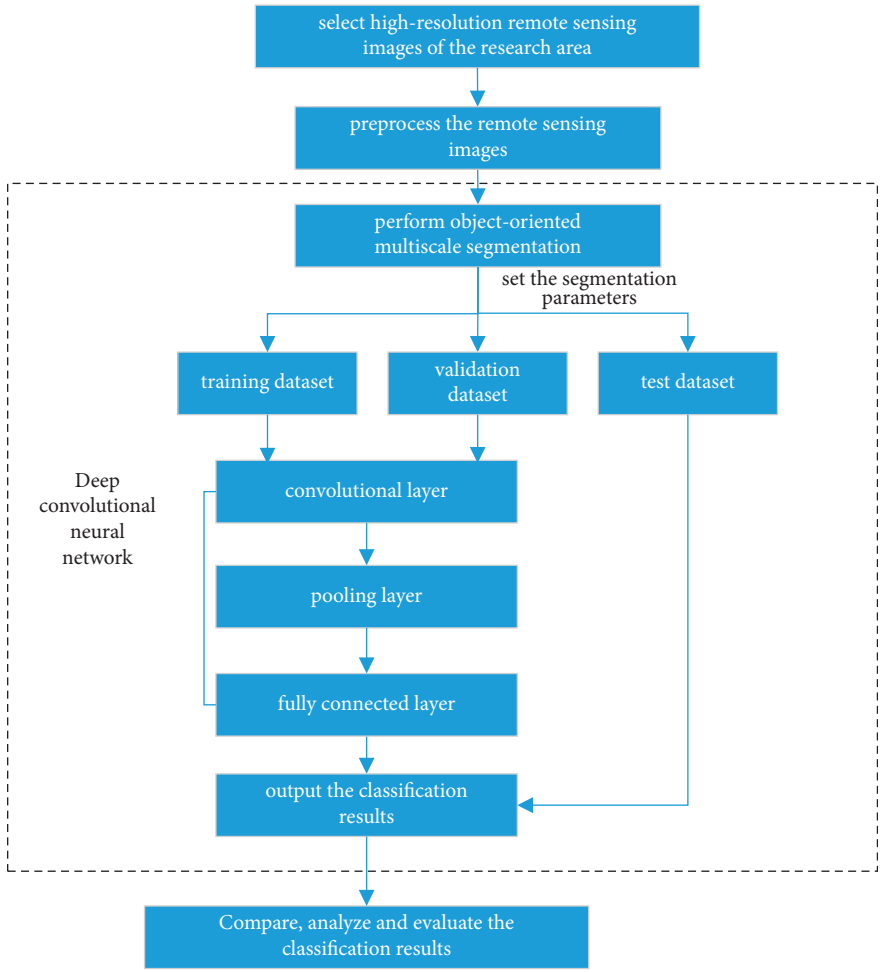


Figure 1: Technology guidelines.

ecosystem with rich biodiversity formed between the interaction between water and land, and they have important ecological value [6, 7]. The status and dynamic changes of wetland resources can be accurately studied in a timely manner by using high-precision and high-spatial-resolution remote sensing image classification methods, which are of great significance in wetland research [8]. However, the boundary between wetland types in high-resolution remote sensing images is not obvious; at present, automatic and efficient extraction of wetland type information has been a bottleneck hindering wetland development and change monitoring [9, 10].

The traditional supervised classification algorithm of remote sensing image based on pixel classifies mainly the image according to the ground object spectrum, which depends on the priori knowledge and practical experience [11]. The diversity of spectral information brings challenges to the high-resolution image classification. The traditional classification method based on physical model and statistical model has been difficult to apply in the remote sensing information extraction in the era of big data [12, 13]. Due to the differences of ground object targets, the neural network needs to extract effectively the features of ground object targets under different receptive fields. Deep learning semantic segmentation based on pixel classification can analyze quickly the deep semantic information of image and has become the most advanced technology in the field of image segmentation [14, 15].

With the improvement of computer technology, the classification method of wetland remote sensing has also been continuously developed. Decision tree has the characteristics of flexibility, intuition, and high efficiency, and it has been widely used in grassland wetland evaluation and freshwater swamp information extraction [16]. In the 1980s, the classification method such as neural network [17] and support vector machine [18] was gradually applied in the classification of wetland remote sensing images. These models can improve the classification accuracy of high-resolution remote sensing images. However, for the complex sample structure, the shallow structure model is affected by the computing power and cannot fully learn the sample information. But the deep structure model can make full use of the spatial structure information of the image by classifying the high-resolution wetland remote sensing image. The convolutional neural network extracts suitably the input features layer by layer from the low level to the high level to form the network weight structure, which is more suitable for dealing with complex ground object classification. More

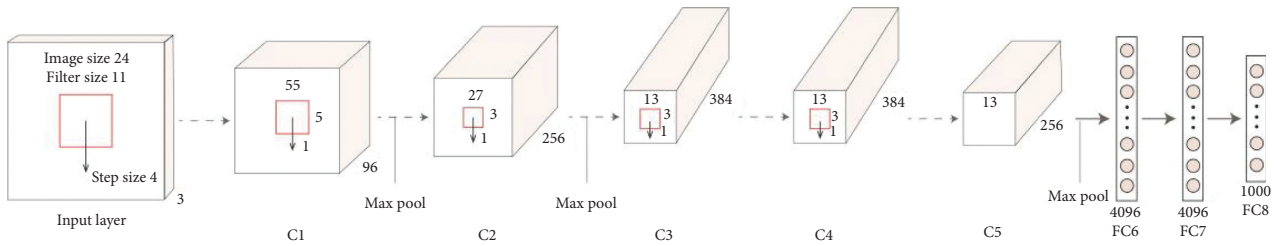


Figure 2: Schematic diagram of AlexNet.

and more scholars introduce convolutional neural network to deal with the problems of feature extraction, classification, and scene recognition of high-resolution remote sensing images, but the research on wetland classification system is very rare.

By outlining classification requirements for extracting wetland type information, this paper presents a method to automatically and efficiently extract wetland type information based on high-spatial-resolution remote sensing images and verifies the classification results by specific experimental cases. By comparing the proposed method with traditional classification methods in terms of extraction precision and efficiency, the high accuracy and efficiency of the method are suggested, and the possibility of accurately and efficiently extracting wetland type information is provided.

2. Basic Idea

Wetlands are considered an important natural resource, and the emergence of high-resolution remote sensing images in recent years has provided a basic condition for large-scale wetland information extraction; by analyzing comprehensively the existing methods, it is possible to improve the accuracy and efficiency. This paper selects an area with rich wetland types as the research area. Based on high-resolution remote sensing image data, the object-oriented method is used to segment the system. A deep convolutional neural network model is used to extract wetland type information, and taking field-measured data as the reference for accuracy verification, the extracted results are compared with traditional classification methods, and an automatic and efficient method for extracting wetland type information is proposed. The method is as follows (Figure 1):

(1) Original high-spatial-resolution remote sensing images in the study area were selected, and the radiometric geometric correction images were preprocessed with methods such as fusion, stitching, and clipping. The distribution characteristics of the remote sensing image features were analyzed by a visual interpretation method, and the wetland type classification system in the study area was established based on the field-measured data.

(2) After preprocessing the remote sensing images for object-oriented multiscale segmentation and standardizing the segmentation of the data, the dataset was divided into a training set, validation set, and test set according to a ratio of 3 : 1 : 1. A deep convolutional neural network was trained based on the depth of the model to achieve the extraction of feature information by the high-resolution remote sensing images.

(3) The iterative self-organizing data analysis technique (ISODATA) algorithm, maximum likelihood, and backpropagation (BP) neural network classification methods were selected to classify the remote sensing images in the study area, and the extraction results of the proposed method were compared with those of the traditional classification methods. In addition, the accuracy of the results of each of the models was evaluated, and the advantages of deep convolutional neural networks for extracting wetland type information from high-resolution remote sensing images were determined.

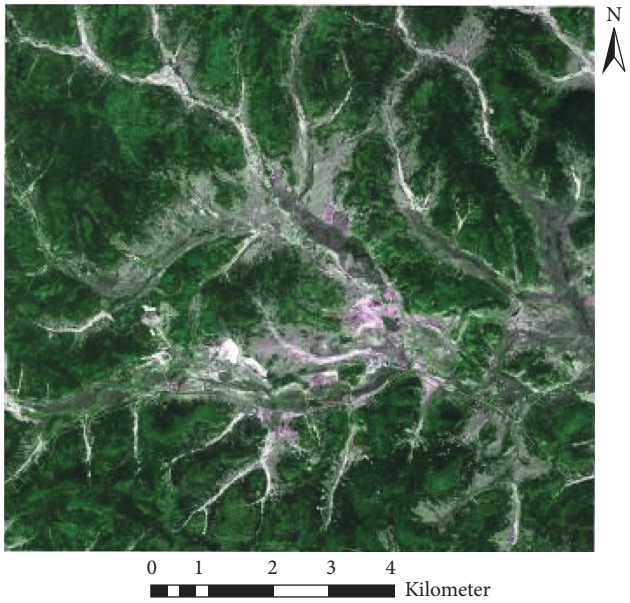


Figure 3: True color image of the study area.

2.1. Object-Oriented Multiscale Segmentation. The object-oriented multiscale remote sensing image segmentation method is a kind of image feature information extraction technology for the classification of high-resolution remote sensing images [19, 20]. Different from traditional remote sensing image classification methods based on pixels, the

Table 1: Multiscale segmentation parameters for extracting wetland type information.

Wetland type	Scale parameter	Shape parameter	Compactness parameter	Smoothness parameter	Weights
Grassland	100	0.4	0.6	0.4	1, 1, 1, 1
Mudflat	100	0.4	0.6	0.4	1, 1, 1, 1
River	30	0.3	0.7	0.3	1, 1, 1, 1

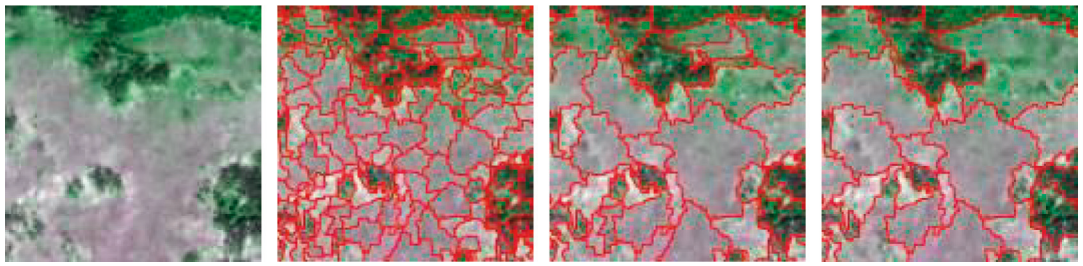


Image 1 Scale=50 Scale=100 Scale=150

Comparison of the multiscale segmentation results of the grassland area

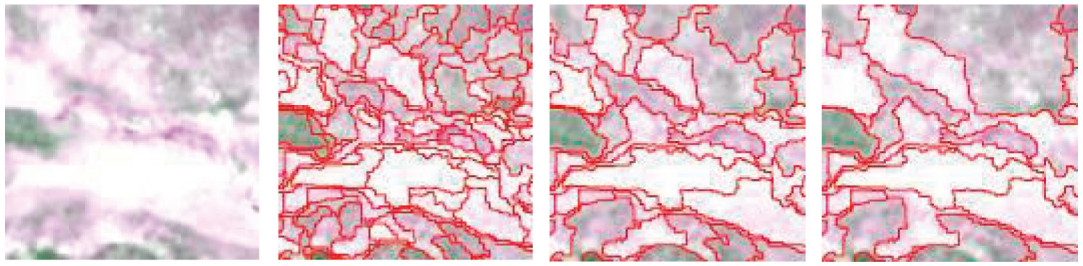


Image 2 Scale=50 Scale=100 Scale=150

Comparison of the multiscale segmentation results for the mudflat area

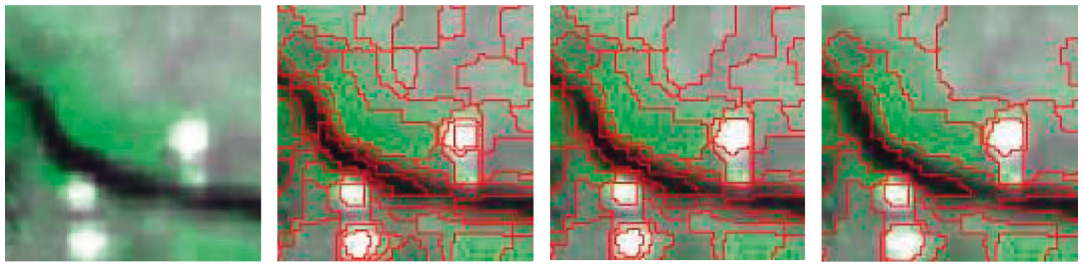


Image 3 Scale=20 Scale=30 Scale=40

Comparison of the multiscale segmentation results for the river

Figure 4: Multiscale segmentation of wetland type information.

object-oriented method divides images into objects according to certain homogeneity and spectral texture structures [21]. The smallest unit is no longer a pixel but an image object. In the segmentation process, similar pixels are divided into the same object, and adjacent objects have obvious differences. The object-oriented method can obtain segmentation objects with relatively regular edges with high spatial accuracy, and the classification results have a certain integrity, which can avoid the occurrence of salt-and-pepper noise [22].

Multiscale segmentation technology is often used in the classification of remote sensing images, and the image segmentation effect is directly affected by the scale parameters. For example, if a large area of image objects is obtained, a large-scale parameter is set, and the number of objects is small correspondingly. When the scale parameters and compactness parameters are fixed, different shape parameters have different effects on the segmentation results. The larger the shape parameter, the more regularly segmented the object; the smaller, the more irregular. In the multiscale segmentation of remote sensing images, the selection of segmentation scale and corresponding segmentation parameters should be based on the image feature. The segmentation scale and parameters corresponding to different feature are different [23]. In this paper, the segmentation scale and corresponding parameters are selected through repeated experiments.

2.2. AlexNet Convolutional Neural Network. The AlexNet model is a typical deep convolutional neural network model; in recent years, it has achieved breakthroughs in the field of image recognition [24, 25]. The AlexNet model obtains the original image characteristics in the input layer and the filter with the sample characteristics. Through 5 convolutional layers and 3 fully connected layers, the depth model with an 8-layer network structure is obtained, which achieves efficient training and has a stable convergence rate. Compared with AlexNet model, other deep convolutional neural network models, such as VGGNet, GoogLeNet, ResNet, and so on, all have far more layers than AlexNet. With the increase of layers, it will bring huge parameter calculation. With the increase of neural network depth, the accuracy of the model will first rise and be saturated. When the depth continues to increase, the accuracy will decline. Because with the increase of the number of layers, there will be gradient explosion or attenuation, the gradient will become unstable, and the value will be particularly large or small. Therefore, the network performance will become worse and worse. Too many parameters lead to excessive memory consumption and excessive computing resources. Although MobileNet and other methods with less layers have less floating-point calculation, networks at different levels can learn the characteristics of different levels. With the reduction of layers, the ability to mine the detailed information of ground objects is affected, and the classification accuracy is reduced [26]. AlexNet uses ReLU activation function to solve the gradient dispersion problem and improve the training speed when the network is deep; dropout is used in training to avoid overfitting. LRN (local response normalization) is used to enhance the generalization ability of the model and improve the accuracy of training. In addition, based on GPU, AlexNet uses CUDA to accelerate the training of neural network and improve the training efficiency of the network. Figure 2 shows the schematic diagram of the AlexNet model.

3. Case Study

3.1. Selection of the Experimental Data. We selected remote sensing images of the Heilongjiang Gongbiela River National Nature Reserve taken by the Gaofen-2 (GF-2) satellite as the experimental data. Panchromatic and corresponding multispectral imaging data of two scenes were collected in July 2018 to obtain a mosaic remote sensing image with a spatial resolution of 1 m after preprocessing. The data are shown in Figure 3.

3.2. Image Multiscale Segmentation. According to visual interpretation, the main wetland types in the preliminary study area were determined. After repeated comparative analysis during image segmentation, the wetland types were first segmented at multiple scales. The boundary of the segmentation objects is more accurate when the segmentation scale parameters of the grassland and mudflat areas (layer 1) are 100. The boundary of the river (layer 2) is more accurate when the segmentation scale parameter is 30. On the condition that the segmentation scales of the above two layers remain unchanged, the weight of the compactness parameters is fixed, and the weight is gradually increased from 0.1 to 0.9 according to an arithmetic sequence to determine the shape parameters. Based on this experiment, layer 1 achieves a better segmentation effect on grassland and mudflat when the segmentation scale parameter is 100, the shape parameter is 0.4, and the compactness parameter is 0.6. Layer 2 is taken as the sublayer of layer 1 and has a better segmentation effect on rivers when the segmentation scale parameter is 30, the shape parameter is 0.3, and the compactness parameter is 0.7. The multiscale segmentation parameters for extracting wetland type information in the study area are shown in Table 1. The multiscale segmentation results are shown in Figure 4.

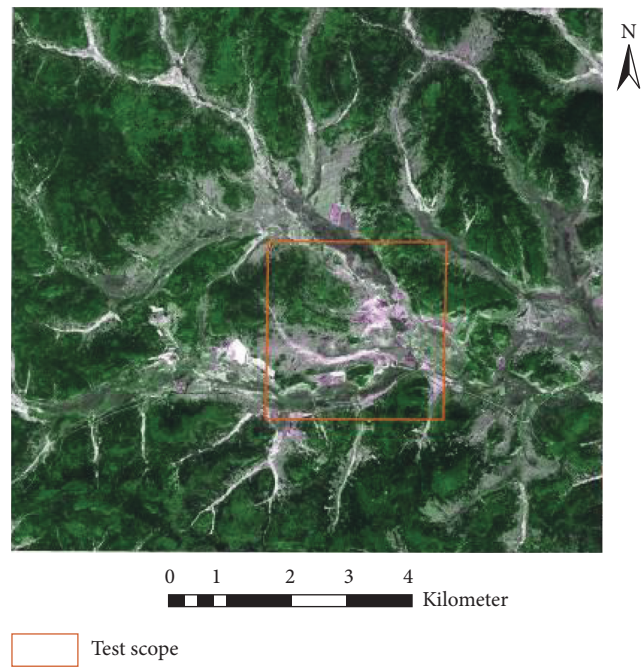


Figure 5: Test scope.

Similarly, the cultivated land area was divided according to a segmentation scale parameter of 100, a shape parameter of 0.4, and a compactness parameter of 0.6. The remaining images to be segmented were forestland (since the road class has a maximum width of 6 pixels and is not considered as a research focus, it is classified as the adjacent feature type).

The segmented wetland images were converted into three-channel red, green, and blue (RGB) images, and the segmented images were divided into three standard datasets of 256 ×× 256 images, namely, a training set, verification set, and test set, according to a ratio of approximately 3 : 1 : 1. A typical area of the research area was selected as the test area, and the segmented images were produced into a standard test set. The scope of the test area is shown in Figure 5. The image segmentation data outside the test area were divided into a standard training set and validation set. There were 3224 samples in the training set, 908 samples in the verification set, and 969 samples in the test set (test areas). Examples of the training set, verification set, and test set are shown in Figure 6. Part of the masked RGB values of the samples are filled in with 0 s.

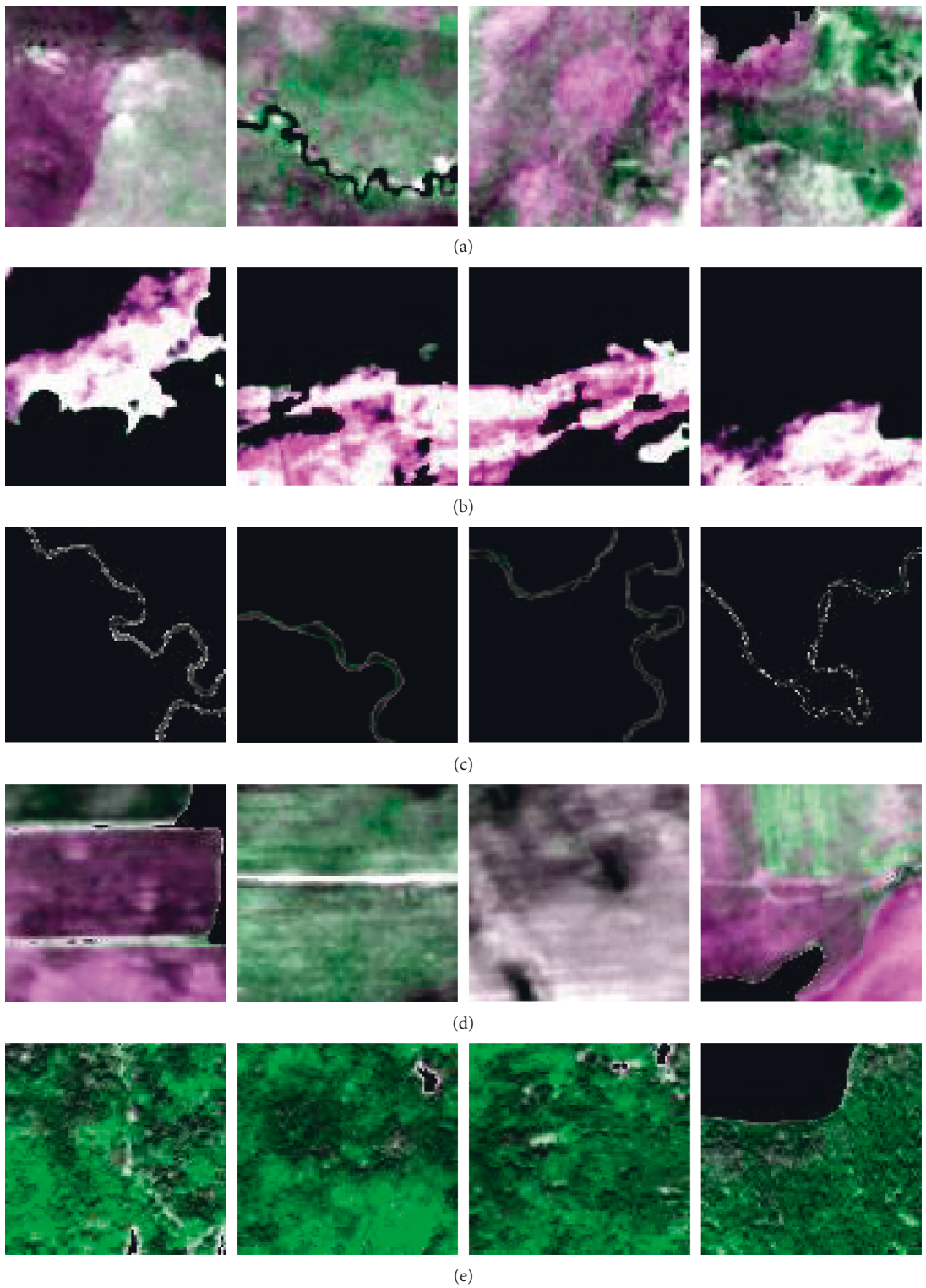


Figure 6: Examples. (a) Grassland. (b) Mudflat. (c) River. (d) Cultivated land. (e) Forest.

3.3. Extraction of the Wetland Type Information. In this study, the Caffe machine learning framework and AlexNet convolutional neural network are applied for wetland type information extraction. During the training process, some neurons were randomly deleted in each iteration, and softmax logistic regression was applied to classify the images in the output layer. The initial value of the learning rate was set to 0.001. The deep convolutional neural network was trained and evaluated through 5 types of land cover types, namely, grassland, mudflat, river, cultivated land, and forest, so there were 5 outputs in the fully connected layer. The network is optimized to obtain the multivariate logistic

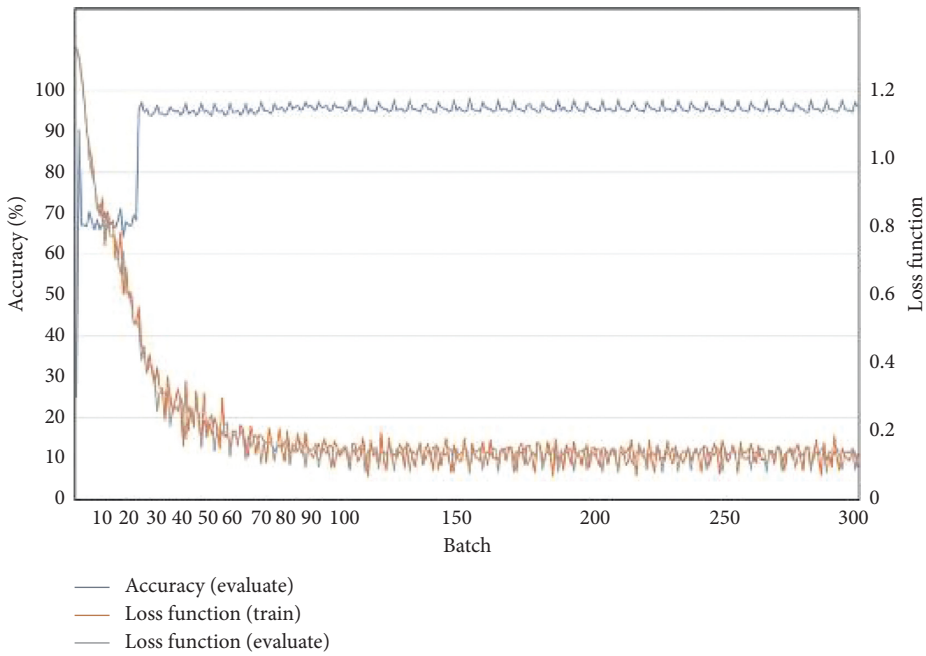


Figure 7: Accuracy and loss function curves.

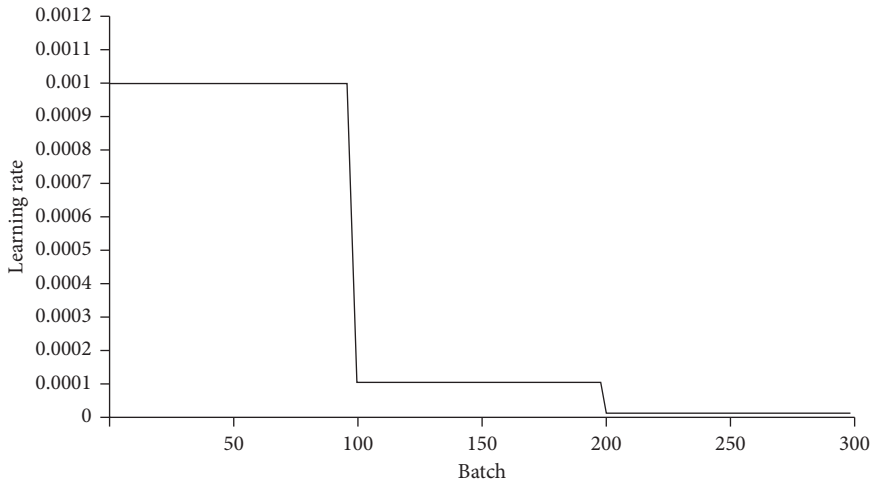


Figure 8: Rate-of-change curve in the training stage.

regression model of the maximum average. The network training process is shown in Figures 7 and 8.

It can be seen from the experiment that the AlexNet deep convolutional neural network model achieves a good effect on the experimental dataset after effective training and learning. The accuracy of batches 0 to 20 was unstable, with an initial accuracy of 25.2%; the accuracy rapidly increased to 90.4%, then rapidly decreased to 67.2%, and stabilized at approximately 70%. After 20 batches, the accuracy began to rise gradually, and the highest accuracy achieved was 97.2%. After 100 batches, the accuracy reached a high level and stabilized. At the same time, the initial loss function was approximately 1.1, and the loss function of batches 0 to 20 rapidly converged to approximately 0.4. The loss function gradually began to converge after 20 batches. The loss function of batch 100 was less than 0.1, and the convergence effect was good and tended to be stable. As the loss function converges rapidly and tends to be stable, the accuracy increases rapidly and remains high. The initial learning rate was 0.001, which gradually decreased to 0.0001 after 100 batches and then continued to decline and gradually stabilized after 200 batches, indicating that the deep convolutional neural network achieves an ideal learning effect on the dataset.

3.4. Analysis and Verification of the Results. This research uses field-measured data and visual interpretation results to verify the effectiveness of the convolutional neural network model for the wetland classification, and its accuracy and efficiency in extracting wetland type information were compared with those of traditional methods, namely, an unsupervised classification method (the ISODATA algorithm), a

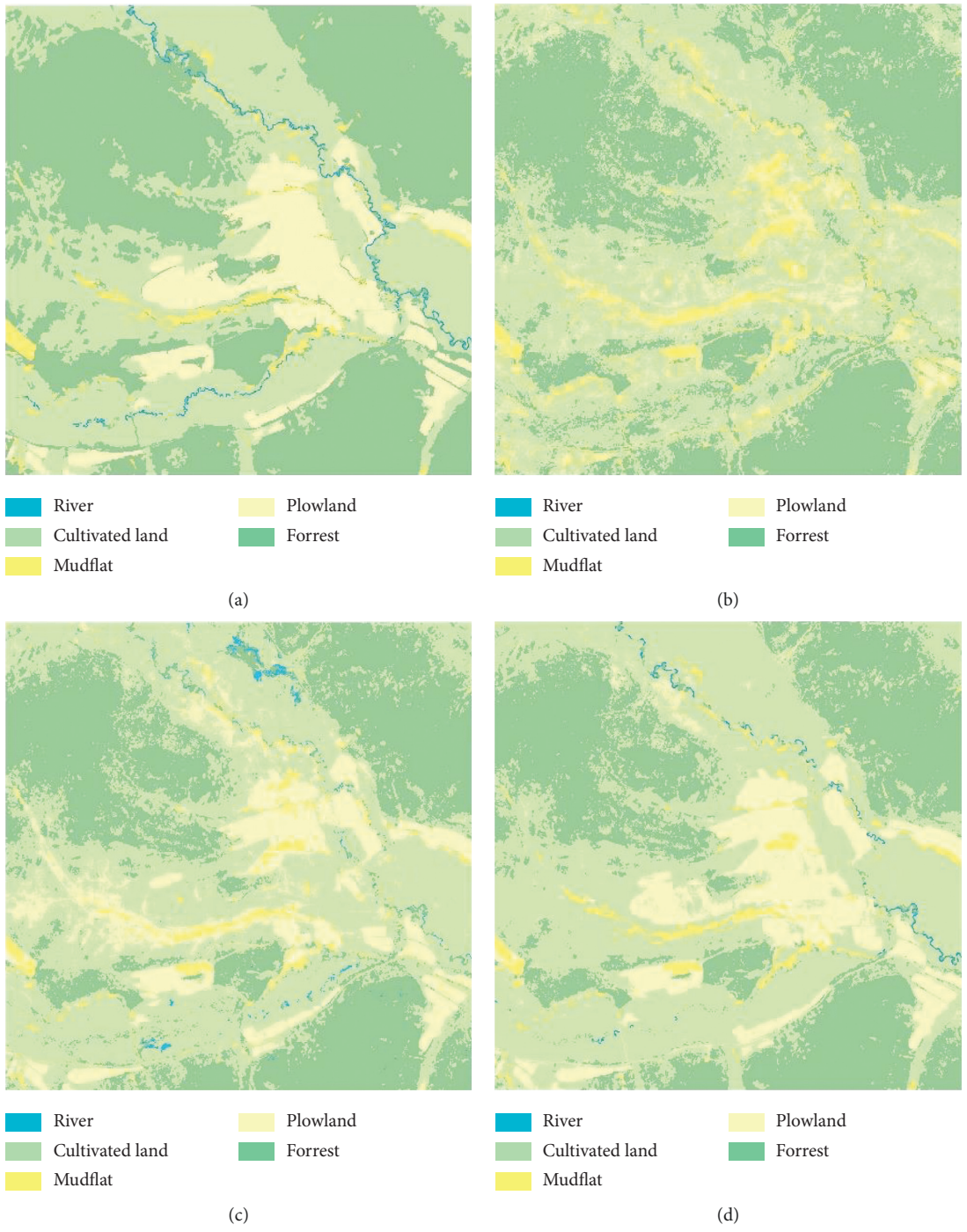


Figure 9: Classification results. (a) Classification results of the method proposed in this paper. (b) Classification results of the ISODATA algorithm. (c) Classification results of the maximum likelihood method. (d) Classification results of the BP neural network.

supervised classification method (the maximum likelihood (ML) method), and shallow machine learning classification method (a BP neural network) on the same data. The classification results of the test set images by the proposed method and the three traditional methods are shown in Figure 9.

All the results of the traditional remote sensing image classification methods, that is, the ISODATA algorithm, ML, and BP neural network, have "salt-and-pepper noise" [27–29]. However, the classification method selected in this paper obtained clear features. After about fifteen repeated experiments, the following indexes are selected to evaluate the accuracy and efficiency of the methods: the producer's accuracy, user's accuracy, overall accuracy, kappa coefficient, and classification time. The classification accuracy and statistical results are shown in Table 2.

Through the comparative analysis of the classification results of the four methods, it can be seen that for the extraction of wetland type information from the high-

Table 2: Comparison of the classification accuracy.

Precision	ISODATA algorithm		Maximum likelihood method		BP neural network		Deep convolutional neural network	
	Producer's accuracy	User's accuracy	Producer's accuracy	User's accuracy	Producer's accuracy	User's accuracy	Producer's accuracy	User's accuracy
River	0.00	0.00	53.23	100.00	66.13	100.00	74.19	100.00
Grassland	87.30	64.45	90.48	69.51	91.01	79.26	95.77	90.95
Mudflat	90.98	93.08	90.23	97.56	91.73	99.19	90.91	99.17
Cultivated land	54.17	49.06	72.92	55.56	87.50	58.33	91.67	65.67
Forest	66.06	68.99	72.12	90.15	86.67	99.31	97.56	98.77
Overall accuracy	70.5193%		80.0670%		87.1022%		92.6050%	
Kappa coefficient	0.5991		0.7336		0.8293		0.9022	
Time (minutes)	26		6		185		4	

resolution remote sensing images, the identification accuracy of the four classification methods is higher for the grassland and mudflat areas, which is due to the large difference in the image characteristics of these land cover types. Although the river image has obvious spectral features, the ISODATA algorithm fails to extract the river information effectively and automatically classifies it into the surrounding features because the local width is no more than 10 pixels. The maximum likelihood classification method can extract the river information, but the river is relatively fragmented. The river features extracted by the BP neural network and deep convolutional neural network are relatively consistent, and the deep convolutional neural network achieves the best river connectivity effect. Compared with the four classification methods, the maximum likelihood method and the deep convolutional neural network method are faster; they take 6 minutes and 4 minutes, respectively. However, the overall classification accuracy of the maximum likelihood method is low. Although the ISODATA algorithm does not need to understand the research area in advance, reduces the interference of human factors, and achieves a moderate classification time (26 minutes), it misclassifies and fails to classify many areas, and its overall classification accuracy is the lowest; therefore, it does not meet the requirements of classification. Although the overall accuracy of the BP neural network classification method is high, it takes 185 minutes to run, which is too long for practical applications. Based on the above analysis, it can be concluded that the deep convolutional neural network can quickly and effectively identify wetland type information, with high classification accuracy, short runtime, and ideal effect.

4. Conclusion and Discussion

At present, the status and dynamic changes of wetland resources cannot be determined in a timely and accurate manner by the current wetland remote sensing method. This paper proposes an automatic method that combines application of object-oriented multiscale segmentation technology and a deep convolutional neural network based on remote sensing image feature extraction technology. This paper extracts wetland type information from high-resolution remote sensing images of the Heilongjiang Gongbiela River National Nature Reserve. The final results are verified and compared with the traditional methods. The results show that the deep convolutional neural network can extract wetland types from remote sensing images better than the traditional methods; for example, the deep convolutional neural network can handle nonobvious boundaries better than the traditional methods. Depth characteristics can be better expressed, and more detailed information of wetland types can be mined, which not only improves the classification accuracy but also substantially improves the extraction efficiency compared with the traditional classification algorithms. The research method proposed in this paper achieves a higher accuracy and has a wider applicability than the traditional methods. However, as a new classification technology, convolutional neural network has broad research space. Based on the results of remote sensing image processing in different regions, this study has still partial limitations and deficiencies, which will be further improved in the future research.

Next, the authors will focus on the GoogLeNet [30], Visual Geometry Group (VGG) [31, 32], and ResNet convolutional neural network architectures and the advantages and disadvantages of neural network models for extracting wetland types from high-resolution remote sensing images. This article created a large number of sample datasets according to the types of features and characteristics of the remote sensing data sources; however, to address a large number of wetland types for environment information extraction from remote sensing images, future research will also increase the number of datasets and diversify the types of datasets, expand the study area, and improve the generalization ability of the model. How to select the optimal parameters and improve the classification efficiency needs the further study.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Guan and Wan constructed the overall framework for this paper. Guan, Wang, and Zhang contributed in the design of case study section. All authors worked collaboratively on writing this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant no. 42001325) and Natural Science Foundation of Anhui Province, China (grant no. 2008085QD168).

References

[1] L. Shi, Y. Wan, X. Gao, and M. Wang, "Feature selection for object-based classification of high-resolution remote sensing images based on the combination of a genetic algorithm and tabu search," *Computational Intelligence and Neuroscience*, vol. 2018, no. 2, Article ID 6595792, 13 pages, 2018.

[2] X. Huang and L. Zhang, "An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 1, pp. 257–272, 2013.

[3] C. Yang, F. Rongshuang, W. Jingxue, W. Zenglin, and S. Ruxing, "High resolution image classification method combining with minimum noise fraction rotation and convolution neural network," *Laser & Optoelectronics Progress*, vol. 54, no. 10, pp. 102801–103445, 2017.

[4] L. Bottou, *Large-scale Kernel Machines*, pp. 321–359, MIT Press, Cambridge, MA, USA, 2007.

[5] L. L. Cao, H. T. Li, Y. S. Han, and H. Liu, "Application of convolutional neural networks in classification of high resolution remote sensing imagery," *Science Surveying and Mapping*, vol. 41, no. 9, pp. 170–175, 2016.

[6] Y. X. Yang, "Main characteristics, progress and prospect of international wetland science research," *Progress in Geography*, vol. 21, no. 2, pp. 111–120, 2002.

[7] S. W. Zhang, F. Q. Yan, L. X. Yu, B. Kun, Y. Jiu-Chun, and C. Li-Ping, "Application of remote sensing technology to wetland research," *Scientia Geographica Sinica*, vol. 33, no. 11, pp. 1406–1412, 2013.

[8] X. Meng, S. Zhang, and S. Zang, "Remote sensing classification of wetland communities based on convolutional neural networks and high resolution images: a case study of the honghe wetland," *Scientia Geographica Sinica*, vol. 38, no. 11, pp. 1914–1923, 2018.

[9] Y. Dang, J. X. Zhang, K. Z. Deng, Y. Zhao, and F. Yu, "Study on the evaluation of land cover classification using remote sensing images based on AlexNet," *Journal of Geo-Information Science*, vol. 19, no. 11, pp. 1530–1537, 2017.

[10] W. Zhang, K. Zheng, P. Tang, and L. J. Zhao, "Land cover classification with features extracted by deep convolutional neural network," *Journal of Image and Graphics*, vol. 22, no. 8, pp. 1144–1153, 2017.

[11] C. Zhang, I. Sargent, X. Pan et al., "Joint deep learning for land cover and land use classification," *Remote Sensing of Environment*, vol. 221, pp. 173–187, 2019.

[12] Y. F. Xi, L. L. Sun, and L. M. He, "Semantic segmentation based on improved deeplab V3+ network," *Computer Systems & Applications*, vol. 29, no. 9, pp. 178–183, 2020.

[13] Y. J. Zheng, S. C. Liu, Q. Du, H. Zhao, X. H. Tong, and M. Dalponte, "A novel multitemporal deep fusion network (MDFN) for short-term multitemporal HR images classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 10691–10704, 2021.

[14] S. Saha, L. Mou, C. Qiu, X. X. Zhu, F. Bovolo, and L. Bruzzone, "Unsupervised deep joint segmentation of multitemporal high-resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 12, pp. 8780–8792, 2020.

[15] X. Y. Tong, G. S. Xia, Q. K. Lu et al., "Land-cover classification with high-resolution remote sensing images using transferable deep models," *Remote Sensing of Environment*, vol. 237, Article ID 111322, 2020.

[16] L. Yu, L. H. He, Q. Zhang, and X. L Wang, "Landsat-TM data based study on dynamic changes of the typical wetlands of poyang lake," *Remote Sensing Information*, vol. 112, no. 6, pp. 48–54, 2010.

[17] Y. Bao and J. Ren, "Wetland landscape classification based on the BP neural network in DaLinor Lake Area," *Procedia Environmental Sciences*, vol. 10, pp. 2360–2366, 2011.

[18] R. Sadeghi, R. Zarkami, K. Sabetraftar, and P. Van Damme, "Use of support vec tor machines (SVMs) to predict distribution of an invasive water fern *Azolla filiculoides* (Lam.) in Anzali wetland, southern Caspian Sea, Iran," *Ecological Modelling*, vol. 244, pp. 117–126, 2012.

[19] H. Huang and K. Xu, "Combing triple-part features of convolutional neural networks for scene classification in remote sensing," *Remote Sensing*, vol. 11, no. 14, p. 1687, 2019.

[20] Z. P. Sun, W. M. Shen, B. Wei et al., "Object-oriented land cover classification using HJ-1 remote sensing imagery," *Science China Earth Sciences*, vol. 53, no. S1, pp. 34–44, 2010.

[21] Z. Zhong, L. Jin, and Z. Xie, "High performance offline handwritten Chinese character recognition using Googlenet and directional feature maps," in *Proceedings of the 13th International Conference on*, pp. 846–850, IEEE, Tunis, Tunisia, August 2015.

[22] W. Su, J. Li, Y. H. Chen, J. S. Zhang, D. Y. Hu, and C. M. Liu, "Object-oriented urban land-cover classification of multi-scale image segmentation method," *Journal of Remote Sensing*, vol. 11, no. 4, pp. 521–530, 2007.

[23] C. Tao, Y. H. Tan, H. J. Cai, D. Bo, and T. Jinwen, "Object-oriented method of hierarchical urban building extraction from high-resolution remote-sensing imagery," *Acta Geodaetica et Cartographica Sinica*, vol. 39, no. 1, pp. 39–45, 2010.

[24] Y. Zhong, J. Zhao, and L. Zhang, "A hybrid object-oriented conditional random field classification framework for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7023–7037, 2014.

[25] L. Y. Feng, *Research on Construction Land Information Extraction from High Resolution Images with Deep Learning Technology*, Zhe Jiang University, Hangzhou, China, 2017.

[26] Z. P. Sun, W. M. Shen, B. Wei et al., "Object-oriented land cover classification using HJ-1 remote sensing imagery," *Science China Earth Sciences*, vol. 53, pp. 34–44, 2010.

[27] D. Browne, M. Giering, and S. Prestwich, "PulseNetOne: fast unsupervised pruning of convolutional neural networks for remote sensing," *Remote Sensing*, vol. 12, no. 7, p. 1092, 2020.
[28] F. Hu, G. S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.
[29] J. Chen, C. Wang, Z. Ma, D. He, and S. Ackland, "Remote sensing scene classification based on convolutional neural networks pre-trained using attention-guided sparse filters," *Remote Sensing*, vol. 10, no. 2, p. 290, 2018.
[30] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-Mobile Net models," *Mobile Information Systems*, vol. 2020, Article ID 7602384, 8 pages, 2020.
[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," pp. 1409–1556, 2014, https://arxiv.org/abs/1409.1556.
[32] X. Wang, L. C. Sui, M. Q. Zhong, D. Li, and L. Dang, "Fully convolution neural networks for water extraction of remote sensing images," *Bulletin of Surveying and Mapping*, vol. 6, no. 6, pp. 41–45, 2018.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4925416, 23 pages
https://doi.org/10.1155/2022/4925416

Review Article

Improved Sparrow Algorithm Based on Game Predatory Mechanism and Suicide Mechanism

Ping Yang,[1] **Shaoqiang Yan**,[1] **Donglin Zhu**,[2] **Jiangpeng Wang**,[1] **Fengxuan Wu**,[1] **Zhe Yan**,[1] **and Song Yan**[1]

[1]*Xi'an Research Institute of High Technology, Xi'an, Shaanxi 710025, China*
[2]*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China*

Correspondence should be addressed to Shaoqiang Yan; yanshaoqiang668@163.com

Received 18 January 2022; Revised 21 March 2022; Accepted 22 April 2022; Published 16 May 2022

Academic Editor: Seyed Jalaleddin Mousavirad

In order to overcome the defect that sparrow search algorithm converges very fast but is easy to fall into the trap of local optimization, based on the original mechanism of sparrow algorithm, this paper proposes game predatory mechanism and suicide mechanism, which makes sparrow algorithm more in line with its biological characteristics and enhances the ability of the algorithm to get rid of the attraction of local optimization while retaining the advantages of fast convergence speed. By initializing the population with the good point set strategy, the quality of the initial population is guaranteed and the diversity of the population is enhanced. In view of the current situation that the diversity index evaluation does not consider the invalid search caused by individuals beyond the boundary in the search process, an index to measure the invalid search beyond the boundary in the search process is proposed, and the measurement of diversity index is further improved to make it more accurate. The improved algorithm is tested on six basic functions and CEC2017 test function to verify its effectiveness. Finally, the improved algorithm is applied to the three-dimensional path planning of UAV with threat area. The results show that the improved algorithm has stronger optimization performance, has strong competitiveness compared with other algorithms, and can quickly plan the effective and stable path of UAV, which improves an effective method for the application in this field and other fields.

1. Introduction

With the progress of modern technology, intelligent algorithms also continue to develop [1–5] and are applied to all kinds of engineering applications and real life. Comparing the swarm intelligence optimization algorithm and traditional optimization algorithm, the former is a heuristic search technology based on biological population characteristics. It has fast convergence speed, strong robustness, and high stability. With its own self-organization and adaptive characteristics, it can effectively solve complex optimization problems. The continuous development of swarm intelligence algorithm not only opens up a new world for solving various complex engineering problems but also stimulates the great research interest of scholars. Various swarm intelligence optimization algorithms appear one after another, and the family of swarm intelligence optimization algorithms has been unprecedentedly developed [6–9]. New algorithms are constantly proposed, such as naked mole-rat algorithm (NMRA) [10], firefly algorithm (FA) [11], ant lion optimizer (ALO) [12], whale optimization algorithm (WOA) [13], sine cosine algorithm (SCA) [14], crow search algorithm (CSA) [15], Harris hawks optimization (HHO) [16], slime mould algorithm (SMA) [17], hunger games search (HGS) [18], Runge–Kutta method [19], and colony predation algorithm (CPA) [20].

Inspired by sparrows' foraging behavior, predatory behavior, and antipredation behavior, a new swarm intelligence optimization algorithm, sparrow search algorithm (SSA) [21], was proposed by Xue et al. in 2020. Compared with traditional intelligent optimization algorithms, such as bat algorithm (BA) [22], grey wolf optimization (GWO) [23], and whale optimization algorithm (WOA) [13], SSA has the advantages of fast convergence, high stability, and strong robustness [24].

The algorithm has the advantage of fast convergence speed, but the global search ability is poor, the optimization results have great randomness, and it is easy to fall into the trap of local optimization. In view of this defect, many scholars have proposed different improvement strategies to improve the performance of sparrow algorithm and successfully solve many complex engineering problems [25]. In literature [26], the search mechanism of bird swarm algorithm replaces the original search mechanism, enhances the global search ability, and effectively breaks through the local limited search. Literature [27] makes full use of the current dominant individuals through the iterative local search mechanism, making the search method more diversified and the optimization accuracy more detailed. Literature [28] introduced the dimension-by-dimension lens learning mechanism to reduce the interference between dimensions and accelerate the convergence of the population. Inspired by logistic model, literature [29] proposed a new adaptive factor to dynamically control the safety threshold, which balances the ability of global search and local development of the algorithm.

The above algorithm has improved the sparrow algorithm and achieved some results, but there are still some shortcomings:

(1) The improved initialization method still has some randomness, which cannot guarantee the quality of each initialization population.

(2) The mechanism that the follower only pillages food with the discoverer's optimal solution does not accord with the characteristics of biological population and makes insufficient use of other optimal solutions, so it is easy to skip the global optimal solution and miss it.

(3) The location update method of followers is to directly jump to the vicinity of the current optimal solution. Although this method leads to the advantage of fast convergence speed, it is also very easy to fall into local optimization. Once it cannot jump out of local optimization, it will reduce the optimization performance.

To solve the above problems, based on the summary of previous work, this paper presents an improved sparrow search algorithm based on game predatory mechanism and suicide mechanism (GPSSA), which helps to improve the shortcomings of SSA algorithm, such as fast convergence speed while being easy to fall into local optimization. This paper presents an improved sparrow search algorithm based on game predatory mechanism and suicide mechanism, which is more in line with the biological habits of sparrow population, ensures more uniform population through a good point set, makes full use of the better individuals in the population through game predatory mechanism and ensures the flexibility of search, and helps the algorithm jump out of local optimum through suicide mechanism. An invalid search index is presented for the true location of individuals beyond the boundary during the search process, which improves the defect of the original index and makes it more accurate and realistic to reflect the population diversity, exploration, and development stages in the search process. The improved GPSSA has the advantages of good initialization population quality, flexible search ability, good population diversity, and fast convergence speed. It is easy to get rid of the characteristics of local optimal attraction. The main contributions are as follows:

(1) By adding good point set, the initial population is more uniform, the population is more diverse, and the quality of the initial population is guaranteed.

(2) On the basis of the original mechanism, a game predatory mechanism is proposed, which makes the algorithm more in line with biological characteristics and makes the search ability of the algorithm more flexible without reducing the advantages of fast convergence speed of the original algorithm.

(3) According to the unique biological characteristics of sparrows that cannot be kept in captivity, a suicide mechanism is proposed, which is to die and live afterwards, eliminate the original individuals and produce new individuals, help the algorithm increase the population diversity, and jump out of the local optimum.

(4) On the basis that the original population diversity index does not consider the individuals beyond the boundary in the search process, an index to measure the invalid search of individuals beyond the boundary is proposed, and the original population diversity index is improved to make the results more accurate.

(5) Six basic test functions are used for experimental simulation. The step size diagram, diversity index analysis, and development exploration stage index are used to analyze the role and effect of each strategy. The improved algorithm is compared with particle swarm optimization (PSO) [30], differential evolution (DE) [31], grey wolf optimization (GWO) algorithm [23], and sparrow search algorithm (SSA) [21] to verify its applicability.

(6) CEC2017 [32, 33] is used to verify the effect of GPSSA algorithm in more complex computing environment. On the basis of the above algorithm, newer cuckoo search (CS) algorithm [34] and multistrategy serial cuckoo search (MSSCS) algorithm [35] are added for comparison and analysis to verify its superiority, and Wilcoxon rank-sum analysis, box chart, and radar sorting chart are used to verify the effectiveness of the algorithm.

(7) The improved algorithm (GPSSA) is applied to a discrete problem, UAV path planning with threat [36], to help UAV quickly plan the optimal path.

2. Sparrow Search Algorithm

SSA is a new swarm intelligence optimization algorithm proposed by sparrows' foraging behavior, predatory behavior, and antipredatory behavior. Its bionic principle is as follows.

When the sparrow population is foraging, the individual population can be divided into discoverer, follower, and scouter. The discoverer is responsible for finding the food and leading the population to search the direction. The follower follows the discoverer to seize the food. The scouter is alert to the threat of the surrounding environment and sends a danger signal in time to remind the sparrow population to move to a safer area.

When the sparrow population does not find the existence of predators or other external threats, the search and foraging environment is safe. The discoverer can perform a wide range of search operations to guide the population to obtain higher energy. When the scouter discovers the threat of the external environment or the presence of predators, with the release of the scouter's early warning signal, the discoverer timely adjusts the search strategy and quickly approaches the safe area. Therefore, the location update formula for the discoverer to guide the population foraging is set as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\dfrac{-i}{\alpha \cdot M}\right), & R_2 < \mathrm{ST}, \\ X_{i,j}^{t} + Q \cdot L, & R_2 \geq \mathrm{ST}. \end{cases} \tag{1}$$

In the above formula, $X_{i,j}$ represents the current position of the i-th sparrow in the j-th dimension. M is the maximum number of iterations. t represents the current number of iterations. R_2 represents an early warning value belonging to [0, 1]. α is a random number belonging to [0, 1]. ST is the security threshold belonging to [0.5,1]. L stands for a $1 \times d$ matrix, and all elements in the matrix are 1. Q is a random number subject to normal distribution.

When the followers follow the discoverers without food, they are very hungry and have low energy, so they need to fly to other places to find food to improve their energy. When the followers follow the discoverer to get food, they only need to find a place near the best position of the sparrow population for foraging. Based on this, the location update formula of the participant is set as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{\mathrm{worst}}^{t} - X_{i,j}^{t}}{i^2}\right), & i > \dfrac{n}{2}, \\ X_{P}^{t+1} + \left|X_{i,j}^{t} - X_{P}^{t+1}\right| \cdot A^{+} \cdot L, & \text{otherwise}. \end{cases} \tag{2}$$

In the above formula, X_p is the best position currently occupied by the discoverer, and X_{worst} represents the worst position. A is a $1 \times d$ matrix with only 1 or -1 elements, where $A^{+} = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}$.

When $i > n/2$, it means that the i-th follower with low fitness does not get food, is in a very hungry state, and needs to fly to other places to find food. When $i \leq n/2$, followers will compete for food with the discoverer who finds the most food (with the best fitness), so as to improve their energy.

When the scouter is at the edge of sparrow population, it is very vulnerable to predator attack or other external environment threats, so the scouter needs to quickly move closer to the global optimal position to reduce the threat. When the scouters are in the middle of the population, if they are aware of the danger information, they will timely get close to other sparrows, so as to reduce the risk of being attacked or preyed on. Therefore, the scouter position update formula is set as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{\mathrm{best}}^{t} + \beta \cdot \left|X_{i,j}^{t} - X_{\mathrm{best}}^{t}\right|, & f_i \neq f_g, \\ X_{i,j}^{t} + K \cdot \left(\dfrac{\left|X_{i,j}^{t} - X_{\mathrm{worst}}^{t}\right|}{(f_i - f_w) + \varepsilon}\right), & f_i = f_g. \end{cases} \tag{3}$$

X_{best} represents the global optimal position of the current population. β is responsible for controlling the step size parameter, which is a random number subject to standard normal distribution. K is responsible for controlling the moving direction and step of sparrow, which is a random number belonging to [−1, 1]. f_i is the fitness value of the current i-th sparrow individual. f_g and f_w, respectively, represent the best and worst fitness values of the current population. ε is a minimal real number.

3. Improved Sparrow Algorithm Based on Game Predatory Mechanism and Suicide Mechanism (GPSSA)

3.1. Good Point Set. The good point set was proposed by Chinese mathematicians Hua and Wang [37], and its principle is as follows: let G_S be the unit cube in s-dimensional Euclidean space, and if $r \in G_s$, the form is

$$P_n(k) = \left\{\left(\left\{r_1^{(n)} \cdot k\right\}, \left\{r_2^{(n)} \cdot k\right\}, \ldots \left\{r_s^{(n)} \cdot k\right\}\right), 1 \leq k \leq n\right\}. \tag{4}$$

If the deviation $\varphi(n)$ satisfies $\varphi(n) = C(r,\varepsilon)n^{-1+\varepsilon}$, ε is any positive number, where $C(r,\varepsilon)n^{-1+\varepsilon}$ is a constant only related to r and ε. $P_n(k)$ is a good point set and r is a good point. $\{r_s^{(n)} \cdot k\}$ represents the decimal part, n represents the number of points, and $r = \{2\cos(2\pi k/p), 1 \leq k \leq s\}$, where p is the minimum prime number satisfying $(p-3)/2 \geq s$. Map it to the search space [34] as

$$x_i(j) = \left(ub_j - lb_j\right) \cdot \left\{r_j^{(i)} \cdot k\right\} + lb_j, \tag{5}$$

where ub_j and lb_j represent the upper and lower bounds of the j-th dimension, respectively.

Figure 1(a) shows a randomly generated initial population distribution map when the good point set is in [0, 1], the number of populations is 500, and the dimension is 1. Figure 1(b)shows the frequency distribution histogram of good point set and tent chaotic map [35] under the above conditions. It can be seen that the good point integration is evenly distributed, and the initialization effect is better than that of tent chaotic map. In the initialization process, the good point set has a more uniform population, which can increase the diversity of the population and help to eliminate the attraction of the local optimal solution.

3.2. Game Predatory Mechanism. The followers in sparrow algorithm have a predatory mechanism; that is, the followers

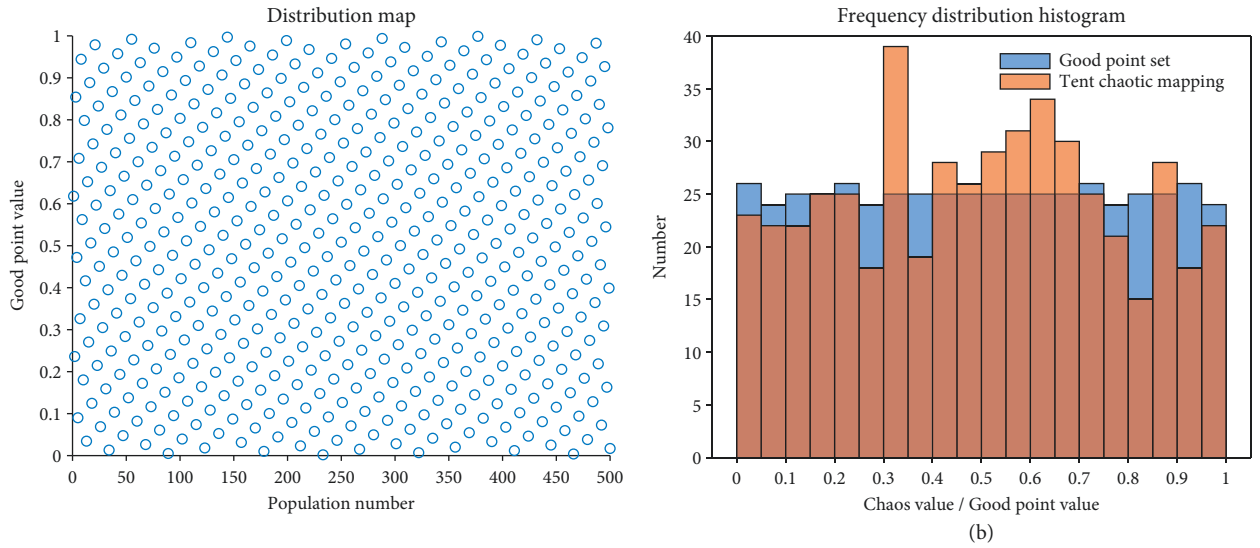


Figure 1: Distribution of good point set.

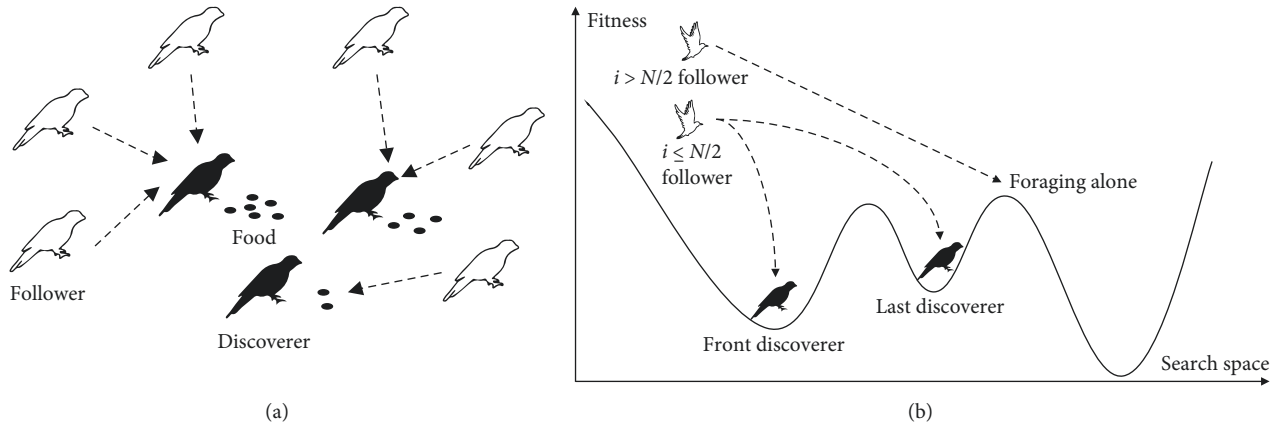


Figure 2: Principle of game grabbing mechanism.

directly jump to the vicinity of the optimal location of the discoverer to compete for food. However, the food in the best position of the discoverer is not endless, and it is impossible to meet the food needs of a large number of followers, which is also inconsistent with the laws of nature. Therefore, based on the game theory, this paper proposes a game predator mechanism; that is, the predatory mechanism of followers and discoverers is regarded as a game process. When $i < N/2$, it is stipulated that the food found by each discoverer can only meet the needs of limited participants, and the amount of food found is determined by its fitness value. The top followers will give priority to those who find the most food to grab food. In order to obtain food and avoid hunger, the lower ranked participants will choose the discoverer who finds less food to eat, as shown in Figure 2(a). It is worth noting that, for the mechanism when $i > N/2$, the followers obtain food and forage alone is retained, as shown in Figure 2(b).

It is found from (2) that when $i \leq N/2$, the followers will grab food after the discoverers. Therefore, there are R followers who directly jump to the vicinity of the discoverer who finds the most food and grab food, where $R = N/2 - \mathrm{PD} \cdot N$ and PD is the proportion of discoverers.

Assuming that the food found by each discoverer is limited, the food in the optimal location of the discoverer cannot meet the food needs of a large number of followers, and the followers participating in the predator mechanism will grab food from different discoverers because of the shortage of food. At this time, it is a game process, and the followers will choose the discoverer who is most favorable to them to grab food. There are two steps in the game predatory mechanism: one is to count the food quantity of the discoverer, and the other is to make game selection to select the most beneficial discoverer for oneself as follows.

3.2.1. Food Quantity Statistics of Discoverer. Individuals with lower fitness are considered to be better choices. The lower the fitness of the discoverer's location is, the more food

is found, and the number of discoverers is $\mathrm{PD} \cdot N$. D_z represents the z-th discoverer. Then the quantity G of food found by the z-th discoverer can be expressed as

$$G_z = \frac{1}{f(D_z)}, \quad z = 1, 2, \ldots, \mathrm{PD} \cdot N. \tag{6}$$

3.2.2. Game Choice. When J_z followers compete for food with the z-th discoverer, the subsequent followers will choose to compete for food with the $z+1$-th discoverer because of the game mechanism, and so on. F_j represents the j-th follower. J_z is determined by the ratio p_z of the quantity G_z of food found by the z-th discoverer to the total quantity. The specific principle is as follows:

$$p_z = \frac{G_z}{\sum_{z=1}^{\mathrm{PD} \cdot N} G_z}, \tag{7}$$

$$J_z = \mathrm{round}(p_z \cdot R). \tag{8}$$

At this time, the relationship between each follower in the game predatory mechanism and its predatory discoverer is as follows:

$$D_1 \begin{Bmatrix} F_1 \\ F_2 \\ \cdots \\ F_{J_1} \end{Bmatrix}, D_2 \begin{Bmatrix} F_{J_1+1} \\ F_{J_1+2} \\ \cdots \\ F_{J_1+J_2} \end{Bmatrix}, \cdots, D_{P\ D \cdot N} \begin{Bmatrix} F_{J_1+J_2+\cdots+J_{\mathrm{PD} \cdot N-1}+1}, \\ F_{J_1+J_2+\cdots+J_{\mathrm{PD} \cdot N-1}+2}, \\ \cdots \\ F_{J_1+J_2+\cdots+J_{\mathrm{PD} \cdot N}}. \end{Bmatrix} \tag{9}$$

Therefore, the location update formula of the improved followers is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{\mathrm{worst}}^t - X_{i,j}^t}{i^2}\right), & i > \dfrac{N}{2}, \\ D_{z,j}^{t+1} + \left|X_{i,j}^t - D_{z,j}^{t+1}\right| \cdot A^+ \cdot L, & \text{otherwise.} \end{cases} \tag{10}$$

3.3. Suicide Mechanism. Sparrows mostly live in places where human beings live. They are very lively, bold, and approachable, but they are very vigilant. Sparrows are very proud birds. They live on people, but they do not want to be kept in captivity. If they do not have freedom, they would rather starve to death [38]. Therefore, this paper adds a suicide mechanism to the original mechanism of the sparrow search algorithm. t_i represents the number of iterations in which the individual's position is not updated. When the individual in the population exceeds the number of dangerous iterations T_c and does not update the position, it is considered that the sparrow is in an imprisoned state. At this time, the sparrow will go on hunger strike and commit suicide. At this time, we abandon the sparrow individual and generate a new sparrow individual, which helps the algorithm jump out of the local optimum. This process can be expressed by an old Chinese saying: "die and come back," and its position is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + X_{i,j}^t \cdot \mathrm{randn}(), & t_i \geq T_c, \\ X_{i,j}^t, & \text{otherwise.} \end{cases} \tag{11}$$

3.4. GPSSA Algorithm Flow

3.5. Time Complexity Analysis. According to the number of populations, iteration times, and dimensions of SSA algorithm, we can easily know that the time complexity of original SSA algorithm is O (P·M·D). Firstly, GPSSA algorithm is improved on the basis of SSA algorithm; GPSSA has the same structure as the original algorithm and does not increase the number of cycles. Secondly, the good point set strategy replaces the original method of randomly initializing the population, but the time complexity does not increase. The game predatory mechanism only changes the way to approach the current optimal solution and further approaches the suboptimal solution to prevent the omission of the suboptimal solution, which also does not increase the time complexity. The suicide mechanism adopts greedy strategy, which increases the complexity of the searcher's algorithm to a certain extent. Only a few individuals in the population increase the complexity of the algorithm but not the order of magnitude of the entire algorithm. To sum up, we can get that the time complexity of GPSSA is still O(P·M·D).

4. Exploration and Exploitation Stage and Population Diversity

Literature [39] proposes a theoretical system to evaluate the algorithm's exploration and exploitation stage and population diversity by using the differences between individual dimensions of the population. On the basis of the above, literature [40] uses the median that can better reflect the population center instead of the average to calculate the dimensional diversity, and the effect is more accurate. The calculation formula is shown in (12). When the population diverges, it indicates that it is in the exploration stage, and the calculation formula is shown in (12). When the population gathers, it indicates that it is in the exploitation stage, and the calculation formula is shown in the following equation:

$$\begin{aligned} \mathrm{Div}_j &= \frac{\sum_{i=1}^N \left|\mathrm{median}(x_j) - x_{i,j}\right|}{N}, \\ \mathrm{Div} &= \frac{\sum_{j=1}^D \mathrm{Div}_j}{D}, \\ X_{\mathrm{pl}} &= \frac{\mathrm{Div}}{\mathrm{Div}_{\max}}, \\ X_{\mathrm{pt}} &= \frac{\left|\mathrm{Div} - \mathrm{Div}_{\max}\right|}{\mathrm{Div}_{\max}}, \end{aligned} \tag{12}$$

where $\mathrm{median}(x_j)$ represents the median of the j-th dimension of the whole population, $x_{i,j}$ represents the j-th dimension of the i-th individual, Div_j represents the diversity of the j-th dimension, Div represents the diversity of

Input:
(1) N: Population sparrows
(2) M: Maximum number of iterations
(3) ST: Alert value
(4) T_c: Number of dangerous iterations
(5) PD: Proportion of discoverers in the population
(6) SD: Proportion of scouters in the population
(7) Output: X_{best}, f_g
(8) $t = 1$;
(9) Initialize population by formula (5).
(10) While ($t < M$)
(11) Sort fitness values and mark the positions of the best and worst sparrows.
(12) $R_2 = \text{rand}(1)$.
(13) For $i = 1$: PD $*$ N
(14) Using formula (1) to update the location of the discoverers;
(15) End for
(16) For $i = (\text{PD} * N + 1)$: N
(17) The discoverers who were going to prey were selected according to formulas (6)–(9);
(18) Using formula (10) to update the location of the followers;
(19) End for
(20) For $l = 1 : \text{SD}$
(21) Using formula (3) to update the location of the scouters;
(22) End for
(23) Update the number of dangerous iterations T_c for each individual;
(24) Update the location of the sparrows according to formula (11);
(25) Get the new optimal individual;
(26) $t = t + 1$;
(27) End while
(28) Return: X_{best}, f_g

Algorithm 1: The framework of the GPSSA.

the whole population, and $\text{Div}_{\max}$ represents the maximum population diversity in the iterative process. In the search process, X_{pl} and X_{pt} represent the percentages of exploration and exploitation stages, respectively.

4.1. Invalid Search Indicator. The above method does not take into account the fact that when an individual exceeds the boundary in the search process, the individual beyond the boundary will be corrected to the boundary in the next iteration. Therefore, the position beyond the boundary is invalid, and the diversity of the population is also invalid or inaccurate. Therefore, this paper proposes an index Inv to evaluate the invalid search beyond the boundary of the algorithm, which is used to describe the ability of invalid search in the t-th iteration of the algorithm. The specific expression is as follows:

$$\text{Inv} = \frac{1}{N \cdot D} \sum_{i=1}^{N} \sum_{j=1}^{D} \max\left(\left|\min\left(X_{i,j}^{t} - lb_j, 0\right)\right|, \left|\min\left(ub_j - X_{i,j}^{t}, 0\right)\right|\right), \quad (13)$$

where $|\min(X_{i,j}^{t} - \text{lb}, 0)|$ and $|\min(\text{ub} - X_{i,j}^{t}, 0)|$ are used to determine the distance values of lb and ub beyond the boundary. If it does not exceed the boundary, the value is 0. $\max(|\min(X_{i,j}^{t} - \text{lb}, 0)|, |\min(\text{ub} - X_{i,j}^{t}, 0)|)$ is used to calculate the distance beyond the boundary. It is easy to know that one of the two must be 0, because it can only exceed one side of the boundary. The smaller the value, the stronger the ability of the algorithm to search effectively.

4.2. Improved Exploration and Exploitation Stage and Population Diversity. If the above indexes for evaluating the invalid search ability are brought into the original index system, more accurate indexes of population diversity and exploration and exploitation stage can be obtained. For individuals beyond the boundary in the search process, the strategy of modifying to the boundary will be adopted in the next iteration, so the boundary position is the real position of individuals beyond the search boundary. Since the individuals beyond the boundary do not affect the selection of the median, the improved formula is as follows:

$$\text{Div}' = \text{Div} - \text{Inv}, \quad (14)$$

$$X_{\text{pl}}' = \frac{\text{Div}'}{\text{Div}_{\max}'}, \quad (15)$$

$$X_{\text{pt}}' = \frac{\left|\text{Div}' - \text{Div}_{\max}'\right|}{\text{Div}_{\max}'}. \quad (16)$$

5. Experimental Simulation of Basic Test Function

In this paper, six basic test functions are selected to verify the performance of GPSSA algorithm and compared with other

Table 1: Parameter settings.

Algorithm	PSO	DE	GWO	CS	MSSCS	SSA	GPSSA
Parameter	$c1 = 2$ $c2 = 2$	CR = 0.2	$a = (2 \longrightarrow 0)$	$\alpha = 0.01$ $\beta = 1.5$	$\alpha = 0.01$ $\beta = 1.5$	SD = 0.2	$SD = 0.2$ PD = 0.2
	$W_{\min} = 0.2$	$F_{\min} = 0.2$		$\beta = 1.5$	Pa = 0.25 $c = 0.2$	PD = 0.2	ST = 0.8
	$W_{\max} = 0.9$	$F_{\max} = 0.8$		$Pa = 0.25$	$PA_{\max} = 0.35$ $PA_{\min} = 0.25$	$ST = 0.8$	$Tc = M/20$

heuristic algorithms, such as PSO, DE, GWO, SSA, and GPSSA. The specific parameter settings are shown in Table 1. F1–F3 are high-dimensional single peak benchmark functions, F4-F5 are high-dimensional multipeak benchmark functions, and F6 is low-dimensional multipeak benchmark function. The test function information is shown in Table 2, and the parameter space is shown in Figure 3.

Figure 4 shows the initial population of the above algorithm and the step size update after only one iteration, in which the objective function is F1 and the number of populations is 100. The red cross in the figure is the location of the optimal solution. As can be seen from the figure, the step amplitude of PSO update is small, and the step amplitude of DE and GWO is large. However, the optimal solution found after initializing the population is not fully utilized by DE and GWO, resulting in blind search. Because of the influence of the discoverer-follower mechanism, most of the population individuals converge near the optimal solution, which is also the main reason why the SSA algorithm has the advantage of fast convergence speed. However, it is also easy to fall into the dilemma of local optimization, and it can be seen from the figure that the initial population of GPSSA is more uniform due to the influence of good point set strategy. Due to the game predatory mechanism, some followers will approach the current suboptimal solution to prevent missing the better solution and falling into the local optimum.

Figure 5 shows the comparison of the effects of the three strategies proposed in this paper. Set the population number to 30 and the number of iterations to 500, and select F1 as the objective function, where (a) is 30 dimensions and (b) is 100 dimensions. SSA is the original algorithm; SSA1, SSA2, and SSA3 are the improved algorithms with good point set strategy, game predatory mechanism, and suicide mechanism separately added to SSA; and GPSSA is the improved algorithm with three strategies added. As can be seen from the above figure, the addition of these three strategies can improve the performance of SSA algorithm to a certain extent. At 30 dimensions, SSA1 and SSA2 are better; at 100 dimensions, the effect of SSA2 and SSA3 is more obvious when the dimension increases, but SSA1 has no significant change. The effect of GPSSA is better than that of a single strategy, which shows that the good point set strategy, game predatory mechanism, and suicide mechanism all have a good improvement effect on the original algorithm.

Table 3 shows the comparison of optimization effects of each algorithm running independently for 30 times, in which the population size is 30, the maximum number of iterations of each algorithm is 500, and the best value of each index is displayed in bold. Finally, the average value is used to sort each algorithm (if the average value is equal, consider the standard deviation). It can be seen from the table that GPSSA shows excellent optimization performance. GPSSA has found the optimal value in F1–F6, indicating that GPSSA has good optimization ability; each index is almost the best in all algorithms, and the comprehensive ranking is also the first, indicating that GPSSA has good optimization accuracy and stability. Except for F5, the results of GPSSA and SSA are the same. In other functions, GPSSA has found a better solution than SSA, indicating that GPSSA algorithm has stronger optimization performance than the original algorithm.

Figure 6 shows the convergence diagram of each algorithm in the above function. GPSSA has faster convergence speed and better optimization accuracy than SSA. In F1 and F2, the convergence rate of GPSSA is very stable, almost in a straight line, and there is no cliff decline like SSA, indicating that the game predatory mechanism has stronger and more stable optimization ability than the original SSA predatory mechanism. In F3 and F4, SSA falls into the local optimum, while GPSSA will regenerate new individuals to jump out of the local optimum after stopping updating due to the suicide mechanism. In F5, although both SSA and GPSSA find the optimal solution, the convergence speed of GPSSA is faster. In F6, both SSA and GPSSA find the optimal solution after falling into the local optimum, which shows that the original SSA algorithm also has a certain ability to jump out of the local optimum, but GPSSA has a stronger ability to jump out of the local optimum.

Figure 7 shows the invalid search index of each algorithm obtained according to formula (13), which is used to represent the index of invalid search caused by exceeding the boundary in the search process. The change trend of this index is mainly related to the update mechanism of the algorithm location. For example, the population will continue to converge in the search process of PSO, DE, and GWO algorithms, so it will continue to decrease with the number of iterations. Although SSA and GPSSA will also converge and converge faster, due to the antipredation behavior in the algorithm, some population individuals will continue to escape the current position and will not be reduced by the number of iterations, such as F3 and F6. F1, F2, F4, and F5 are not affected by the update step size and boundary size.

Because an individual experiences the phenomenon of exceeding the boundary in the search process and the individual is corrected to the boundary in the next iteration update, the boundary position is the real position of the

Table 2: Basic test function.

Function	Dimensions	Interval	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/100	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−100, 100]	0
$F_3(x) = \sum_{i=1}^{n} \mathrm{ix}_i^4 + \mathrm{random}[0\ ,\ 1)$	30	[−1.28, 1.28]	0
$F_4(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500, 500]	−418.98 ∗ dim
$F_5(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
$F_6(x) = \sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.4029

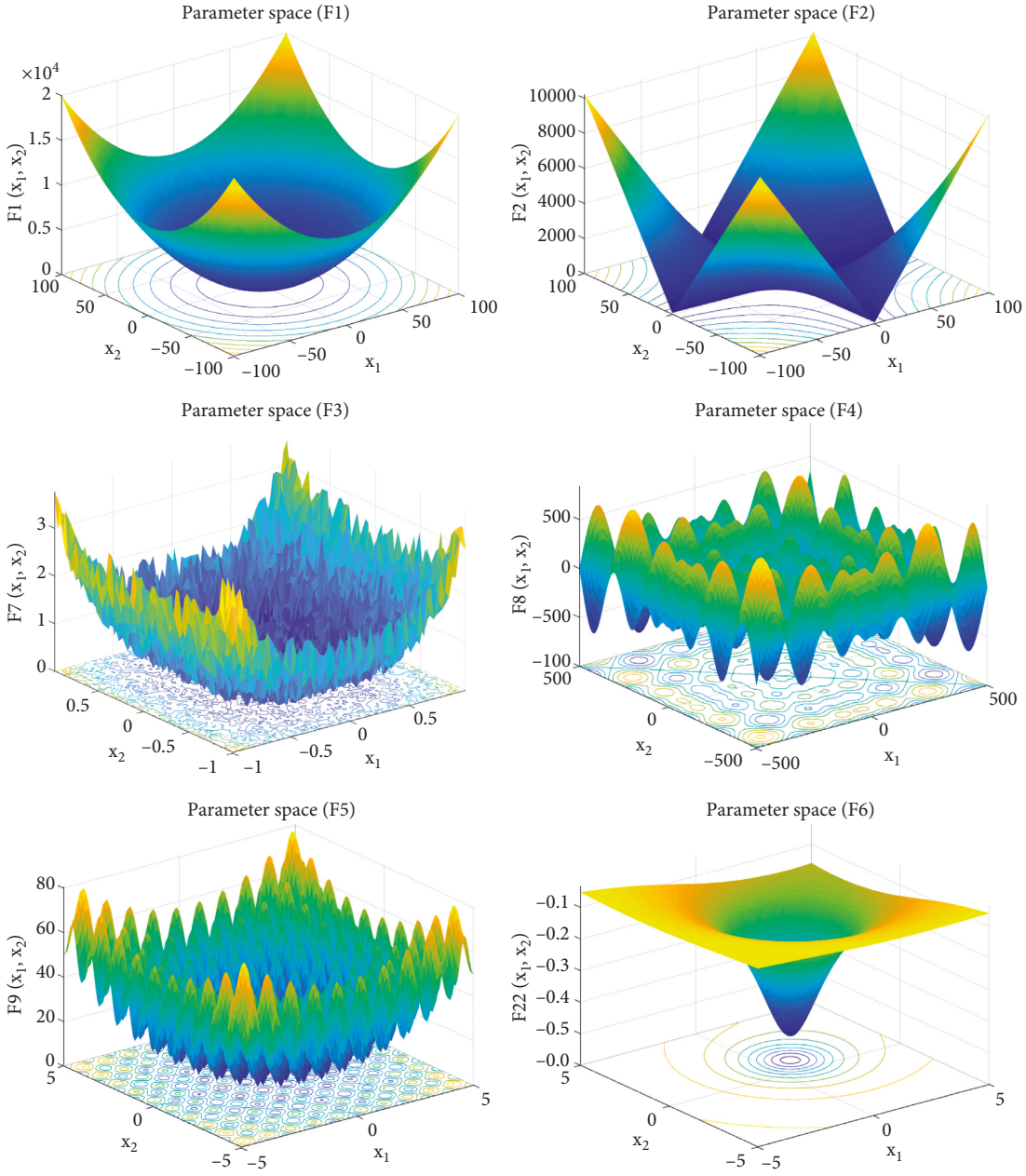


Figure 3: Parameter space of function.

individual exceeding the boundary in the search process. The improved diversity index removes such effects and the results will be more accurate. Figure 8 shows the population diversity calculated according to the improved formula (14). It can be seen from Figure 8 that the diversity of PSO, DE, and GWO algorithms will decrease with the number of iterations due to the continuous convergence of population individuals, and the decline speed of PSO is the slowest.

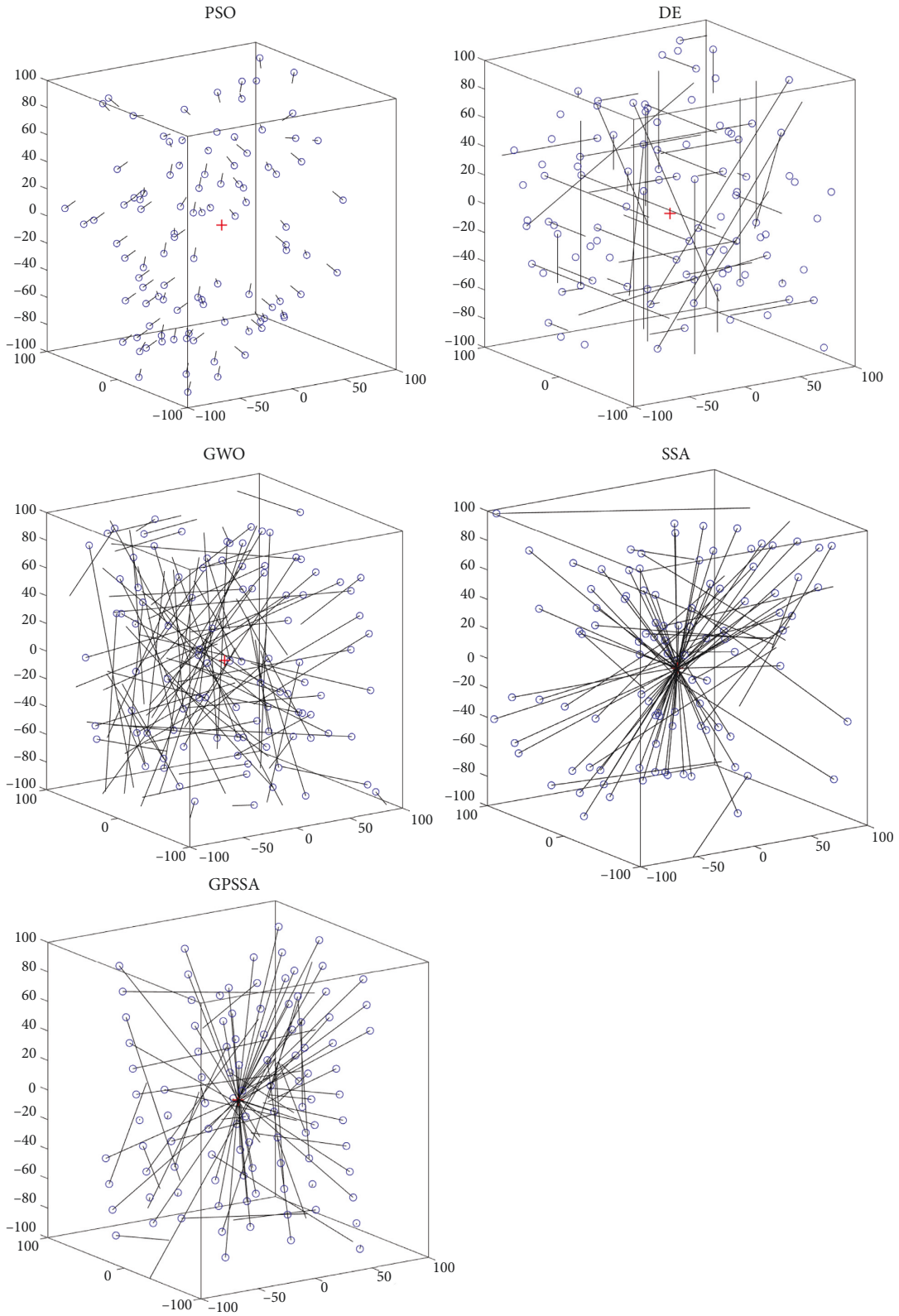


Figure 4: Algorithm step diagram.

SSA and GPSS algorithms decline faster, but because the vigilant mechanism continues to escape from the current position, it will not fall to 0, but it will continue to fluctuate to maintain population diversity and prevent falling into local optimization. Combined with Figure 6, we can find that when SSA and GPSSA no longer converge (when the convergence curve level or no longer decreases), the population diversity will suddenly rise due to the

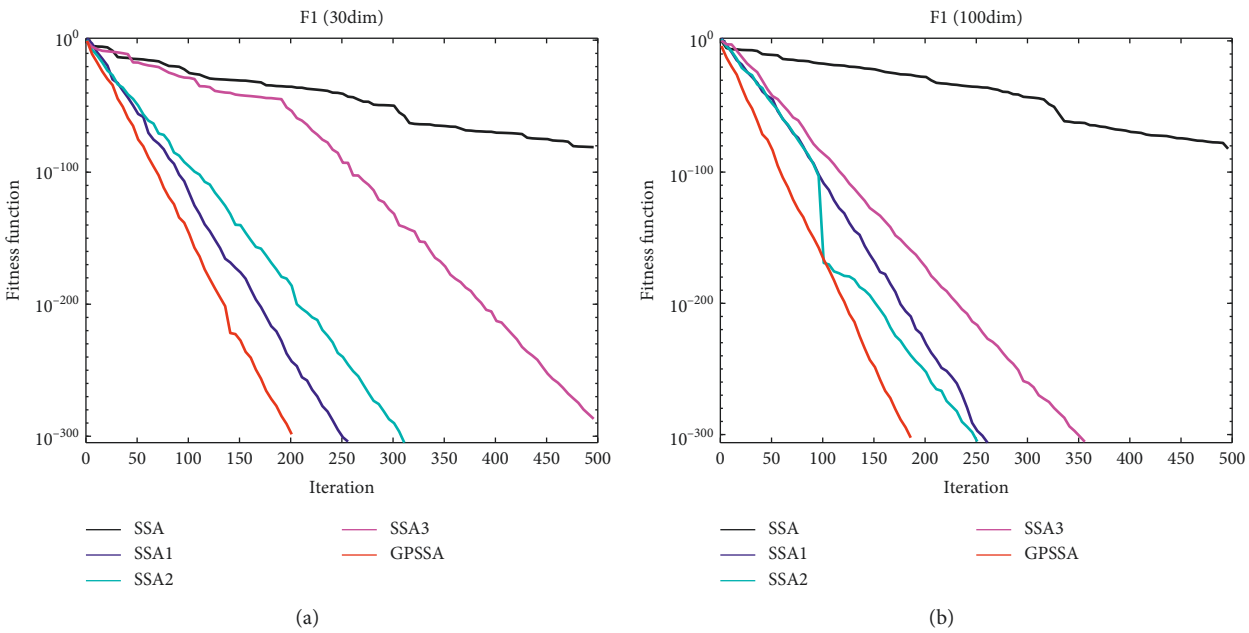


Figure 5: Strategy effect comparison.

Table 3: Comparison table of optimization effect.

	Index	PSO	DE	GWO	SSA	GPSSA
F1	Best	$8.93E-06$	$2.48E+01$	$6.31E-28$	**$0.00E+00$**	**$0.00E+00$**
	Worst	$8.36E-04$	$7.54E+04$	$4.66E-25$	$1.37E-67$	**$0.00E+00$**
	Ave	$1.63E-04$	$3.72E+04$	$5.25E-26$	$4.58E-69$	**$0.00E+00$**
	Std	$2.13E-04$	$2.08E+04$	$1.13E-25$	$2.51E-68$	**$0.00E+00$**
	Rank	4	5	3	2	1
F2	Best	$4.10E-03$	$5.61E+00$	$1.47E-16$	**$0.00E+00$**	**$0.00E+00$**
	Worst	$2.71E-01$	$3.71E+13$	$1.61E-15$	$3.45E-40$	**$0.00E+00$**
	Ave	$4.03E-02$	$1.28E+12$	$6.19E-16$	$1.15E-41$	**$0.00E+00$**
	Std	$5.51E-02$	$6.77E+12$	$3.56E-16$	$6.30E-41$	**$0.00E+00$**
	Rank	4	5	3	2	1
F3	Best	$6.02E-02$	$1.93E-03$	$5.47E-04$	$5.86E-06$	**3.90E − 06**
	Worst	$3.15E-01$	$1.64E+02$	$7.20E-03$	$1.28E-03$	**2.47E − 04**
	Ave	$1.63E-01$	$4.72E+01$	$1.93E-03$	$2.86E-04$	**5.88E − 05**
	Std	$6.39E-02$	$5.11E+01$	$1.52E-03$	$3.05E-04$	**5.21E − 05**
	Rank	4	5	3	2	1
F4	Best	$-7.10E+03$	$-1.13E+04$	$-7.18E+03$	**$-1.26E+04$**	**$-1.26E+04$**
	Worst	$-3.07E+03$	$-1.61E+03$	$-3.05E+03$	$-5.56E+03$	**$-1.03E+04$**
	Ave	$-5.37E+03$	$-4.73E+03$	$-5.62E+03$	$-9.83E+03$	**$-1.21E+04$**
	Std	$1.12E+03$	$2.25E+03$	$1.09E+03$	$2.34E+03$	**$6.38E+02$**
	Rank	4	5	3	2	1
F5	Best	$3.59E+01$	$6.24E+01$	$5.68E-14$	**$0.00E+00$**	**$0.00E+00$**
	Worst	$9.06E+01$	$4.45E+02$	$7.83E+00$	**$0.00E+00$**	**$0.00E+00$**
	Ave	$5.76E+01$	$3.11E+02$	$9.05E-01$	**$0.00E+00$**	**$0.00E+00$**
	Std	$1.30E+01$	$1.25E+02$	$1.97E+00$	**$0.00E+00$**	**$0.00E+00$**
	Rank	4	5	3	1	1
F6	Best	**$-1.04E+01$**	$-6.43E+00$	**$-1.04E+01$**	**$-1.04E+01$**	**$-1.04E+01$**
	Worst	$-2.75E+00$	$-5.28E-01$	$-5.09E+00$	$-5.09E+00$	**$-1.04E+01$**
	Ave	$-8.59E+00$	$-1.76E+00$	$-9.87E+00$	$-9.34E+00$	**$-1.04E+01$**
	Std	$3.11E+00$	$1.36E+00$	$1.62E+00$	$2.16E+00$	**2.53E − 07**
	Rank	4	5	2	3	1
Average rank		4	5	2.83	2	1

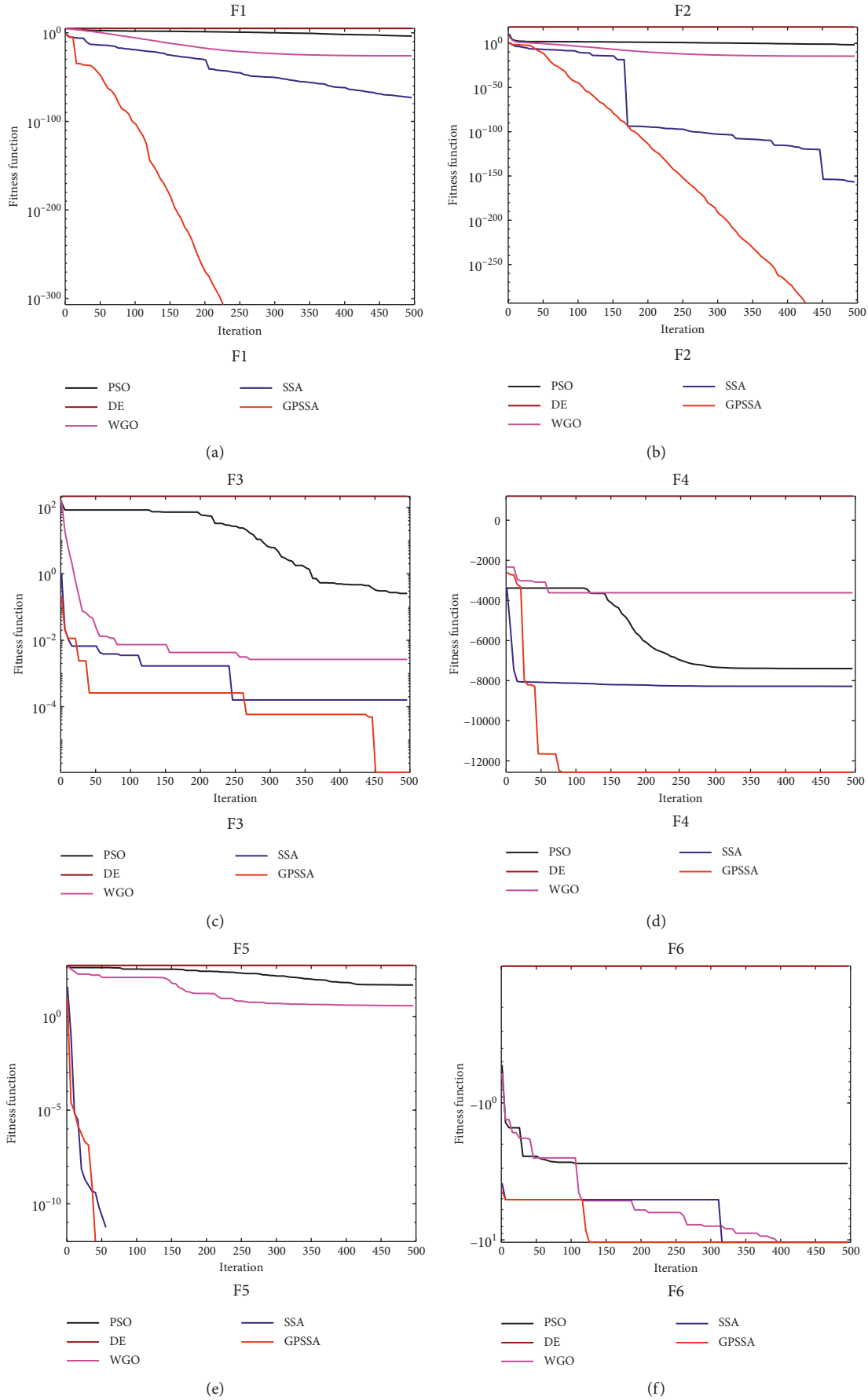


Figure 6: Convergence diagram of each algorithm.

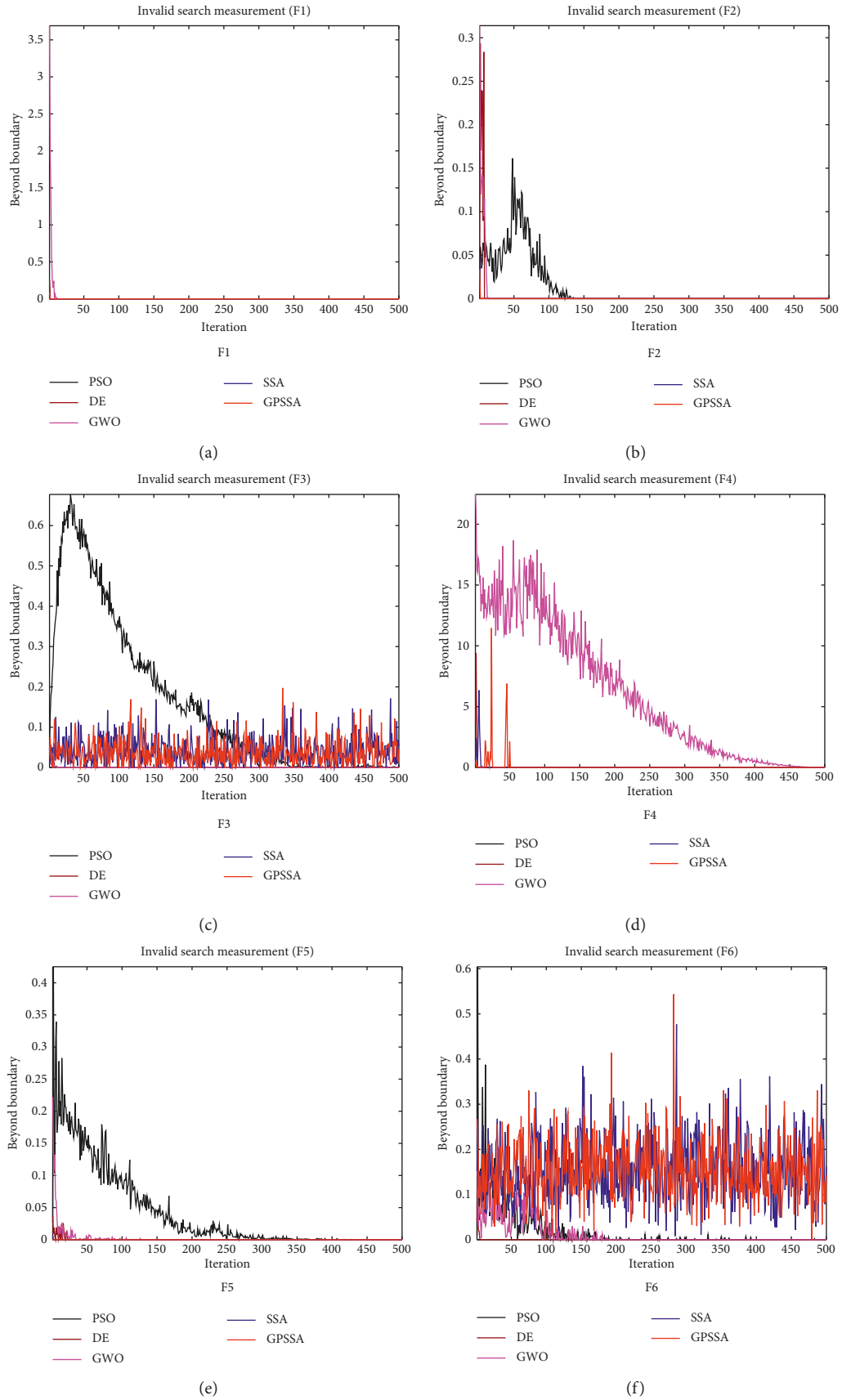


Figure 7: Invalid search of each algorithm.

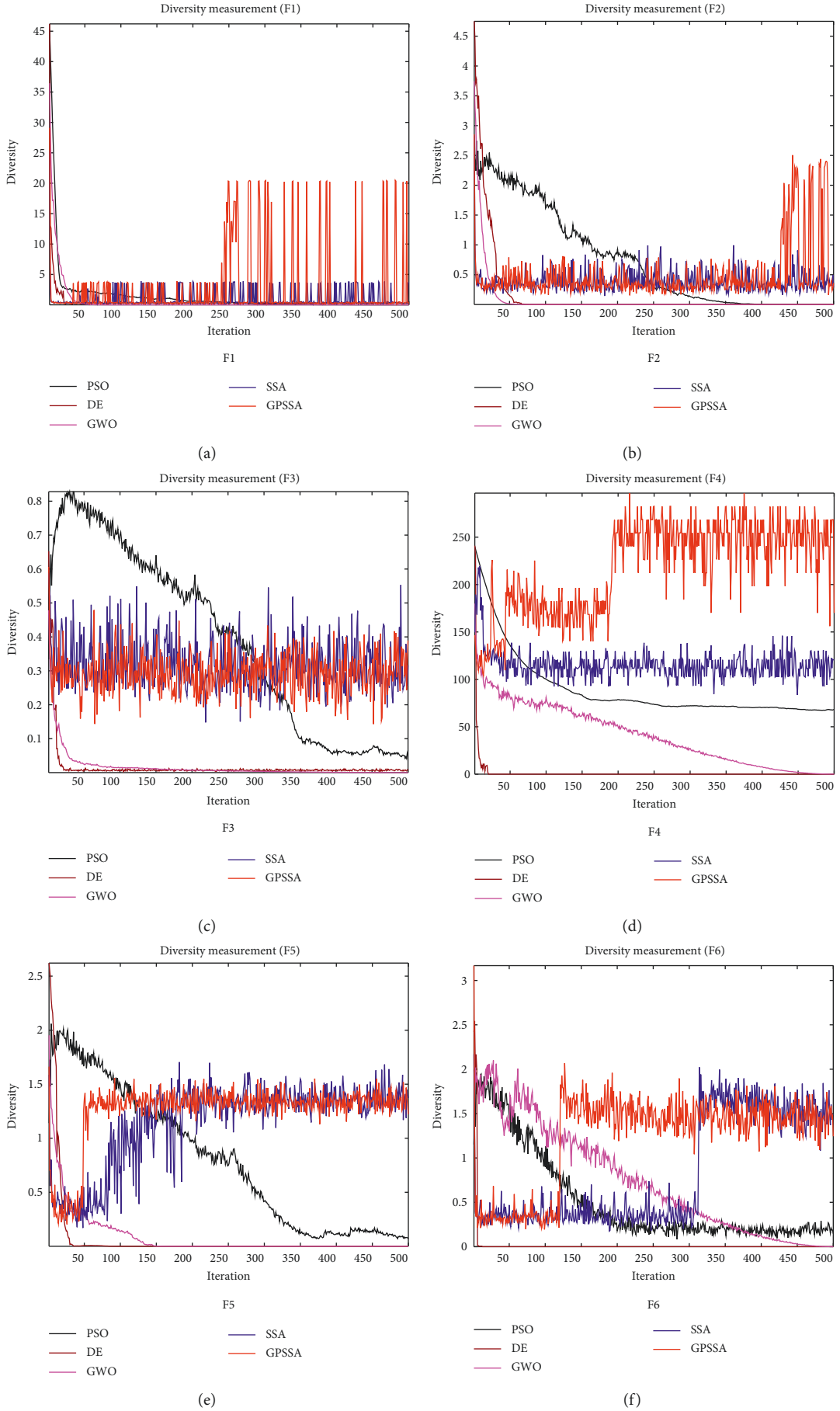


Figure 8: Diversity of each algorithm.

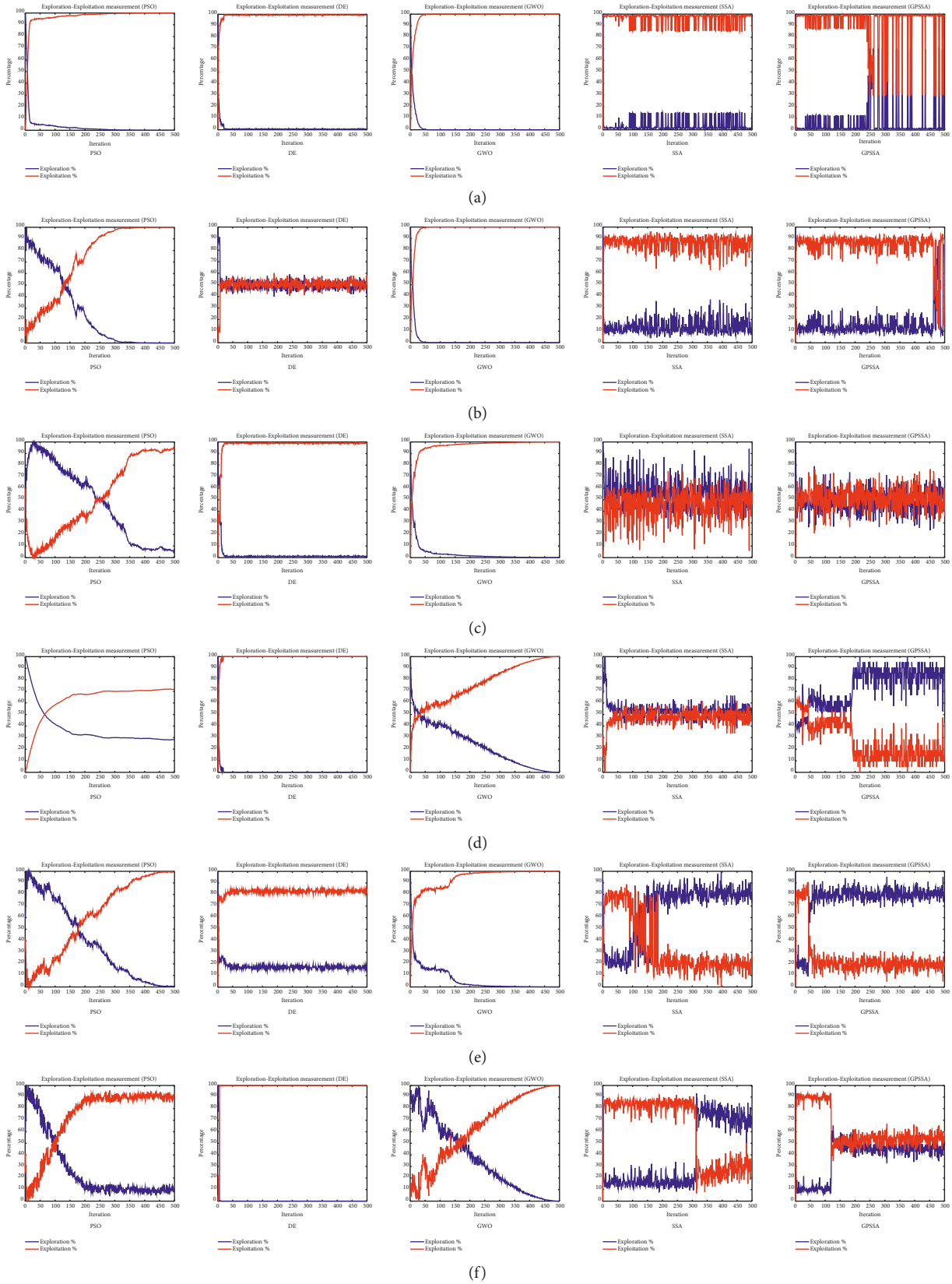


Figure 9: Exploration-exploitation percentage of each algorithm.

Table 4: CEC2017 test results.

	Index	PSO	DE	GWO	CS	MSSCS	SSA	GPSSA
F1	Ave	3.32E + 03	3.44E + 07	8.25E + 08	9.97E + 07	3.24E + 03	3.82E + 03	**1.57E + 03**
	Std	3.40E + 03	2.68E + 06	5.83E + 08	4.18E + 07	3.35E + 03	3.29E + 03	**2.26E + 03**
	Rank	3	5	7	6	2	4	1
F3	Ave	**3.00E + 02**	1.22E + 05	2.82E + 04	3.81E + 04	2.32E + 04	5.03E + 04	2.19E + 04
	Std	**1.03E − 01**	9.82E + 03	9.91E + 03	1.23E + 04	7.21E + 03	2.05E + 04	1.47E + 04
	Rank	1	7	4	5	3	6	2
F4	Ave	**4.73E + 02**	5.41E + 02	5.35E + 02	5.12E + 02	5.06E + 02	5.10E + 02	4.88E + 02
	Std	2.77E + 01	**4.18E + 00**	2.72E + 01	2.62E + 01	2.28E + 01	1.58E + 01	2.90E + 01
	Rank	1	7	6	5	3	4	2
F5	Ave	6.38E + 02	6.82E + 02	8.03E + 02	6.33E + 02	6.14E + 02	7.80E + 02	**5.95E + 02**
	Std	1.30E + 01	1.11E + 01	**7.68E + 00**	1.89E + 01	1.79E + 01	6.20E + 01	3.72E + 01
	Rank	4	5	7	3	2	6	1
F6	Ave	6.38E + 02	**6.02E + 02**	6.57E + 02	6.25E + 02	6.30E + 02	6.65E + 02	6.03E + 02
	Std	9.06E + 00	**1.06E − 01**	6.18E + 00	2.46E − 01	1.66E − 01	7.27E + 00	1.55E + 00
	Rank	5	1	6	3	4	7	2
F7	Ave	**8.30E + 02**	9.28E + 02	1.33E + 03	8.64E + 02	8.69E + 02	1.33E + 03	8.42E + 02
	Std	2.13E + 01	**7.59E + 00**	1.01E + 01	2.09E + 01	1.96E + 01	1.36E + 01	2.72E + 01
	Rank	1	5	7	3	4	6	2
F8	Ave	9.11E + 02	9.85E + 02	9.74E + 02	9.47E + 02	9.29E + 02	9.91E + 02	**8.86E + 02**
	Std	1.58E + 01	**9.28E + 00**	3.22E + 01	1.78E + 01	1.66E + 01	3.13E + 01	2.55E + 01
	Rank	2	6	5	4	3	7	1
F9	Ave	3.08E + 03	2.89E + 03	5.41E + 03	1.05E + 03	**1.00E + 03**	5.42E + 03	1.38E + 03
	Std	5.67E + 02	2.09E + 02	**3.09E + 01**	7.65E + 01	5.66E + 01	1.10E + 02	3.09E + 02
	Rank	5	4	6	2	1	7	3
F10	Ave	**4.23E + 03**	7.21E + 03	5.75E + 03	5.41E + 03	4.86E + 03	6.25E + 03	4.43E + 03
	Std	5.71E + 02	**1.93E + 02**	1.02E + 03	2.80E + 02	4.16E + 02	1.13E + 03	1.12E + 03
	Rank	1	7	5	4	3	6	2
F11	Ave	1.20E + 03	1.62E + 03	1.40E + 03	1.21E + 03	**1.20E + 03**	1.24E + 03	1.22E + 03
	Std	**2.28E + 01**	8.83E + 01	9.95E + 01	2.80E + 01	2.62E + 01	4.38E + 01	4.74E + 01
	Rank	2	7	6	3	1	5	4
F12	Ave	**5.18E + 04**	5.16E + 07	2.78E + 07	8.08E + 09	7.08E + 09	1.92E + 06	1.16E + 06
	Std	**2.60E + 04**	6.84E + 06	2.74E + 07	4.21E + 09	4.93E + 09	1.36E + 06	1.50E + 06
	Rank	1	5	4	7	6	3	2
F13	Ave	9.40E + 03	4.28E + 06	2.48E + 06	7.07E + 08	3.52E + 08	1.49E + 07	**4.56E + 03**
	Std	1.14E + 04	1.35E + 06	1.31E + 07	2.64E + 09	1.92E + 09	5.73E + 07	**2.40E + 03**
	Rank	2	4	3	7	6	5	1
F14	Ave	8.51E + 03	1.17E + 05	2.94E + 05	1.59E + 03	**1.56E + 03**	1.76E + 04	6.20E + 03
	Std	4.65E + 03	5.33E + 04	3.29E + 05	2.51E + 01	**1.28E + 01**	1.44E + 04	1.18E + 03
	Rank	4	6	7	2	1	5	3
F15	Ave	1.61E + 04	3.74E + 05	3.48E + 04	2.15E + 03	**1.73E + 03**	7.62E + 03	3.57E + 03
	Std	1.19E + 04	2.64E + 05	2.32E + 04	1.77E + 02	**3.09E + 01**	6.45E + 03	7.42E + 02
	Rank	5	7	6	2	1	4	3
F16	Ave	2.53E + 03	2.74E + 03	3.37E + 03	2.51E + 03	2.36E + 03	3.57E + 03	**2.36E + 03**
	Std	1.82E + 02	**8.27E + 01**	5.91E + 02	2.51E + 02	2.05E + 02	6.71E + 02	2.63E + 02
	Rank	4	5	6	3	2	7	1
F17	Ave	2.12E + 03	2.01E + 03	2.79E + 03	1.98E + 03	**1.96E + 03**	2.87E + 03	1.98E + 03
	Std	2.07E + 02	**5.48E + 01**	2.84E + 02	9.25E + 01	6.33E + 01	2.51E + 02	1.73E + 02
	Rank	5	4	6	2	1	7	3
F18	Ave	1.45E + 05	1.21E + 06	7.22E + 05	1.17E + 05	**5.71E + 04**	1.25E + 05	3.61E + 05
	Std	1.47E + 05	3.49E + 05	6.37E + 05	4.01E + 04	**2.29E + 04**	2.10E + 05	1.63E + 05
	Rank	4	7	6	2	1	3	5
F19	Ave	9.39E + 03	4.08E + 05	7.29E + 05	2.84E + 03	**2.08E + 03**	6.35E + 03	2.86E + 03
	Std	1.02E + 04	9.24E + 04	2.38E + 06	4.61E + 02	**2.63E + 01**	3.84E + 03	5.71E + 02
	Rank	5	6	7	2	1	4	3

Table 4: Continued.

	Index	PSO	DE	GWO	CS	MSSCS	SSA	GPSSA
	Ave	2.46$E+03$	2.34$E+03$	2.80$E+03$	2.32$E+03$	**2.30$E+03$**	2.72$E+03$	2.37$E+03$
F20	Std	1.18$E+02$	6.38$E+01$	**4.06$E+01$**	9.61$E+01$	8.90$E+01$	2.01$E+02$	1.13$E+02$
	Rank	5	3	7	2	1	6	4
	Ave	2.42$E+03$	2.49$E+03$	2.56$E+03$	2.49$E+03$	2.49$E+03$	2.55$E+03$	**2.38$E+03$**
F21	Std	2.25$E+01$	**7.43$E+00$**	2.71$E+01$	4.14$E+01$	4.97$E+01$	2.46$E+01$	2.20$E+01$
	Rank	2	3	7	5	4	6	1
	Ave	**3.46$E+03$**	4.51$E+03$	6.95$E+03$	3.60$E+03$	3.61$E+03$	6.97$E+03$	4.55$E+03$
F22	Std	1.85$E+03$	**2.43$E+02$**	1.76$E+03$	2.04$E+03$	1.86$E+03$	1.09$E+03$	2.28$E+03$
	Rank	1	4	6	2	3	7	5
	Ave	3.03$E+03$	2.83$E+03$	3.08$E+03$	2.85$E+03$	2.87$E+03$	3.26$E+03$	**2.79$E+03$**
F23	Std	9.38$E+01$	**7.69$E+00$**	8.69$E+01$	3.23$E+01$	2.04$E+01$	1.09$E+02$	7.37$E+01$
	Rank	5	2	6	3	4	7	1
	Ave	3.14$E+03$	3.02$E+03$	3.21$E+03$	3.02$E+03$	3.03$E+03$	3.39$E+03$	**2.91$E+03$**
F24	Std	1.06$E+02$	**6.89$E+00$**	5.94$E+01$	2.12$E+01$	6.03$E+01$	9.44$E+01$	3.89$E+01$
	Rank	5	2	6	3	4	7	1
	Ave	**2.88$E+03$**	2.92$E+03$	2.96$E+03$	3.00$E+03$	3.03$E+03$	2.91$E+03$	2.90$E+03$
F25	Std	8.15$E+00$	5.19$E+00$	2.40$E+01$	1.48$E+00$	**9.70$E-01$**	1.51$E+01$	1.85$E+01$
	Rank	1	4	5	6	7	3	2
	Ave	6.02$E+03$	5.40$E+03$	8.84$E+03$	3.49$E+03$	**3.37$E+03$**	7.63$E+03$	4.45$E+03$
F26	Std	2.17$E+03$	**8.24$E+01$**	9.26$E+02$	8.16$E+02$	6.27$E+02$	1.54$E+03$	3.63$E+02$
	Rank	5	4	7	2	1	6	3
	Ave	**3.21$E+03$**	3.23$E+03$	3.46$E+03$	3.34$E+03$	3.37$E+03$	3.59$E+03$	3.23$E+03$
F27	Std	1.05$E+02$	**3.01$E+00$**	1.17$E+02$	8.91$E+00$	1.01$E+01$	2.75$E+02$	1.86$E+01$
	Rank	1	3	6	4	5	7	2
	Ave	3.19$E+03$	3.33$E+03$	3.36$E+03$	3.36$E+03$	3.37$E+03$	3.24$E+03$	**3.19$E+03$**
F28	Std	5.53$E+01$	**6.87$E+00$**	3.59$E+01$	2.30$E+01$	2.13$E+01$	2.37$E+01$	5.41$E+01$
	Rank	2	4	6	5	7	3	1
	Ave	3.78$E+03$	4.03$E+03$	5.37$E+03$	3.78$E+03$	3.76$E+03$	5.02$E+03$	**3.68$E+03$**
F29	Std	2.43$E+02$	**9.67$E+01$**	9.24$E+02$	1.10$E+02$	1.18$E+02$	5.75$E+02$	1.42$E+02$
	Rank	3	5	7	4	2	6	1
	Ave	**5.26$E+03$**	5.28$E+05$	7.35$E+06$	2.41$E+04$	1.33$E+04$	2.76$E+04$	2.43$E+04$
F30	Std	**2.38$E+03$**	1.63$E+05$	3.03$E+06$	6.52$E+03$	3.18$E+03$	1.18$E+04$	1.02$E+04$
	Rank	1	6	7	3	2	5	4
Average rank	2.97	4.76	6.00	3.59	2.93	5.48	2.28	

antipredation behavior, which helps the algorithm jump out of the local optimum. At the same time, due to the suicide mechanism, GPSSA will think that it is in an imprisoned state at this time and then commit suicide to produce new individuals, which will have better population diversity compared to the SSA algorithm.

According to equations (15) and (16), Figure 9 shows the exploration-exploitation ratio of each algorithm in six functions, where (a), (b), (c), (d), (e), and (f) are the development exploration ratio under F1, F2, F3, F4, F5, and F6, respectively. As can be seen from the figure, PSO, DE, and GWO algorithms all change from exploration stage to exploitation stage with the number of iterations, and the exploitation percentage increases and the exploration percentage decreases. SSA and GPSS algorithms enter the development stage after very few iterations because of their fast convergence speed. Combined with Figure 6, we can find that when SSA and GPSSA are no longer convergent, the development percentage continues to decline, the exploration percentage increases, and the development stage changes to the exploration stage to prevent them from falling into local optimization. Due to the addition of three strategies, GPSSA has better search ability and the ability to jump out of local optimization, which leads GPSSA to find the optimal solution earlier, end the exploitation stage, and then enter the exploration stage.

6. CEC2017 Test

In order to further verify the generality of the algorithm, the algorithm is tested on CEC2017 test function [41]. Due to the defects of F2 function, this paper will not test. The number of dimensions is 30, the number of individuals is 100, the number of evaluations is set to 10000 $*$ dim, and the algorithm parameters remain unchanged. After 30 independent runs, the average value and standard deviation of each algorithm are calculated according to the results and finally sorted. The best value for each index is displayed in bold. The specific test results are shown in Table 4.

The results in Table 4 show that the average value of GPSSA algorithm is the best value among all algorithms in F1, F5, F8, F9, f13, F14, F15, F16, F17, F19, F21, F23, F24, F26, F28, and F29 functions, which shows that GPSSA algorithm has good optimization ability and

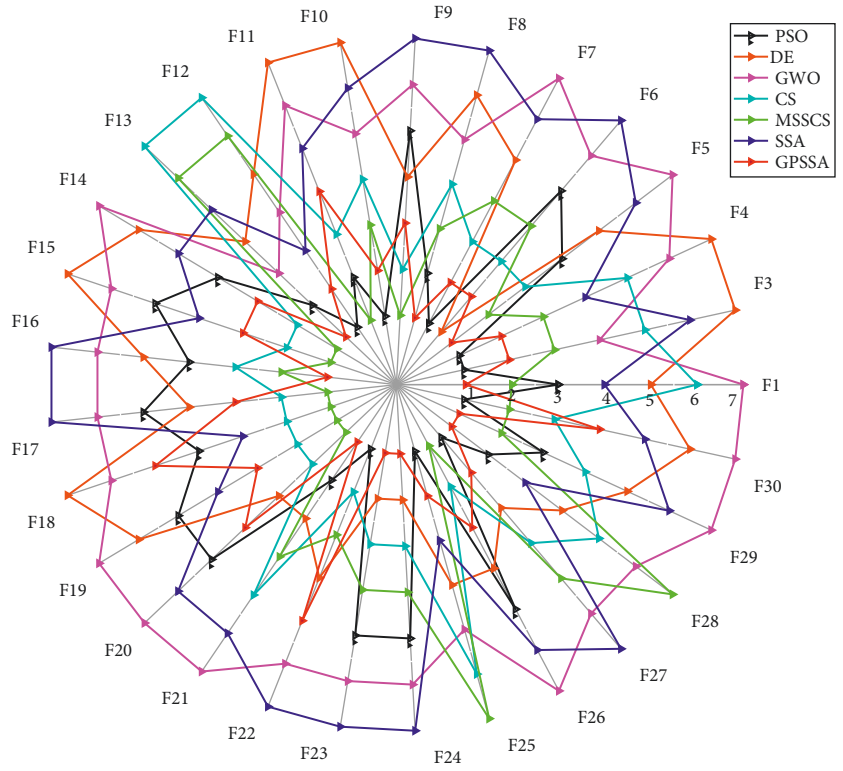


Figure 10: Ranking radar chart.

Table 5: Wilcoxon rank-sum test results.

F	F1	F3	F4	F5	F6	F7	F8	F9	F10	F11
PSO	$2.20E-02$	$2.78E-11$	$4.52E-01$	$1.15E-07$	$2.51E-11$	$6.71E-02$	$8.20E-07$	$3.25E-11$	$6.62E-01$	$7.18E-02$
DE	$2.66E-11$	$2.80E-11$	$7.74E-09$	$8.10E-09$	$2.39E-01$	$2.57E-11$	$2.38E-10$	$2.79E-11$	$7.22E-09$	$2.75E-11$
GWO	$2.69E-11$	$8.97E-03$	$1.90E-07$	$2.44E-11$	$2.51E-11$	$2.73E-11$	$1.63E-10$	$2.76E-11$	$2.10E-06$	$6.20E-10$
CS	$2.65E-11$	$8.15E-06$	$1.25E-04$	$5.09E-07$	$2.88E-11$	$2.99E-03$	$7.59E-10$	$3.02E-06$	$4.20E-08$	$6.30E-01$
MSSCS	$2.67E-02$	$1.58E-01$	$8.61E-04$	$3.33E-04$	$2.88E-11$	$1.14E-04$	$3.60E-09$	$5.75E-09$	$3.82E-03$	$7.21E-03$
SSA	$7.38E-03$	$1.74E-08$	$8.73E-05$	$9.22E-11$	$2.72E-11$	$2.52E-11$	$1.02E-10$	$2.66E-11$	$4.03E-08$	$3.02E-01$
	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
PSO	$2.94E-11$	$5.58E-01$	$5.28E-03$	$1.02E-03$	$3.72E-04$	$4.36E-03$	$2.21E-05$	$2.50E-01$	$1.06E-02$	$1.43E-08$
DE	$2.59E-11$	$2.83E-11$	$2.58E-11$	$2.42E-11$	$5.14E-09$	$7.93E-02$	$4.48E-11$	$2.57E-11$	$6.09E-01$	$7.48E-11$
CS	$6.16E-11$	$2.81E-11$	$3.99E-10$	$2.45E-11$	$2.00E-10$	$2.83E-11$	$6.43E-03$	$2.52E-11$	$2.42E-11$	$2.78E-11$
MSSCS	$2.23E-08$	$6.88E-09$	$2.63E-11$	$4.17E-11$	$1.83E-03$	$3.25E-01$	$1.42E-07$	$8.53E-01$	$1.90E-01$	$1.13E-09$
GWO	$1.40E-04$	$2.60E-03$	$2.63E-11$	$2.52E-11$	$6.84E-01$	$3.95E-01$	$8.08E-09$	$8.13E-10$	$4.34E-02$	$1.51E-08$
SSA	$3.08E-02$	$2.92E-09$	$3.06E-10$	$2.73E-05$	$1.60E-10$	$2.64E-11$	$5.30E-07$	$4.25E-04$	$3.80E-09$	$2.41E-11$
	F22	F23	F24	F25	F26	F27	F28	F29	F30	+/=/−
PSO	$2.52E-04$	$2.74E-11$	$8.73E-11$	$1.06E-06$	$7.61E-03$	$9.31E-08$	$8.65E-01$	$1.44E-01$	$2.41E-11$	21/0/8
DE	$8.19E-01$	$1.85E-01$	$5.22E-10$	$2.94E-06$	$4.81E-10$	$9.70E-01$	$2.78E-11$	$3.07E-10$	$2.78E-11$	23/0/6
GWO	$2.96E-05$	$2.83E-11$	$2.78E-11$	$1.95E-09$	$2.43E-11$	$2.83E-11$	$3.74E-11$	$4.59E-11$	$2.75E-11$	29/0/0
CS	$2.91E-02$	$1.50E-02$	$7.03E-10$	$2.72E-11$	$3.05E-04$	$2.93E-11$	$3.00E-11$	$2.02E-03$	$1.53E-01$	25/0/5
MSSCS	$5.36E-02$	$1.59E-03$	$4.78E-09$	$2.72E-11$	$2.32E-06$	$2.93E-11$	$3.00E-11$	$7.44E-02$	$1.24E-09$	25/0/5
SSA	$6.79E-07$	$2.47E-11$	$2.48E-11$	$3.49E-03$	$4.63E-10$	$2.44E-11$	$9.40E-05$	$2.68E-11$	$1.66E-01$	27/0/2

accuracy. The average ranking of GPSSA is 2.28, followed by PSO (2.97), MSSCS (2.93), CS (3.59), DE (4.76), SSA (5.48), and GWO (6.00). The SSA algorithm does not perform well in more complex CEC217. Only due to GWO, it shows that the optimization ability of the original algorithm decreases in more complex optimization problems, while the improved GPSSA is more suitable for complex optimization problems due to the improvement of mechanism.

Figure 10 shows a radar chart made according to the algorithm ranking. The higher the ranking is, the closer it is to the radar center. It can be seen that more than half of GPSSA functions are closest to the radar center, and the enclosed area is the smallest. According to the 30 times' optimal results, all algorithms conduct Wilcoxon rank-sum

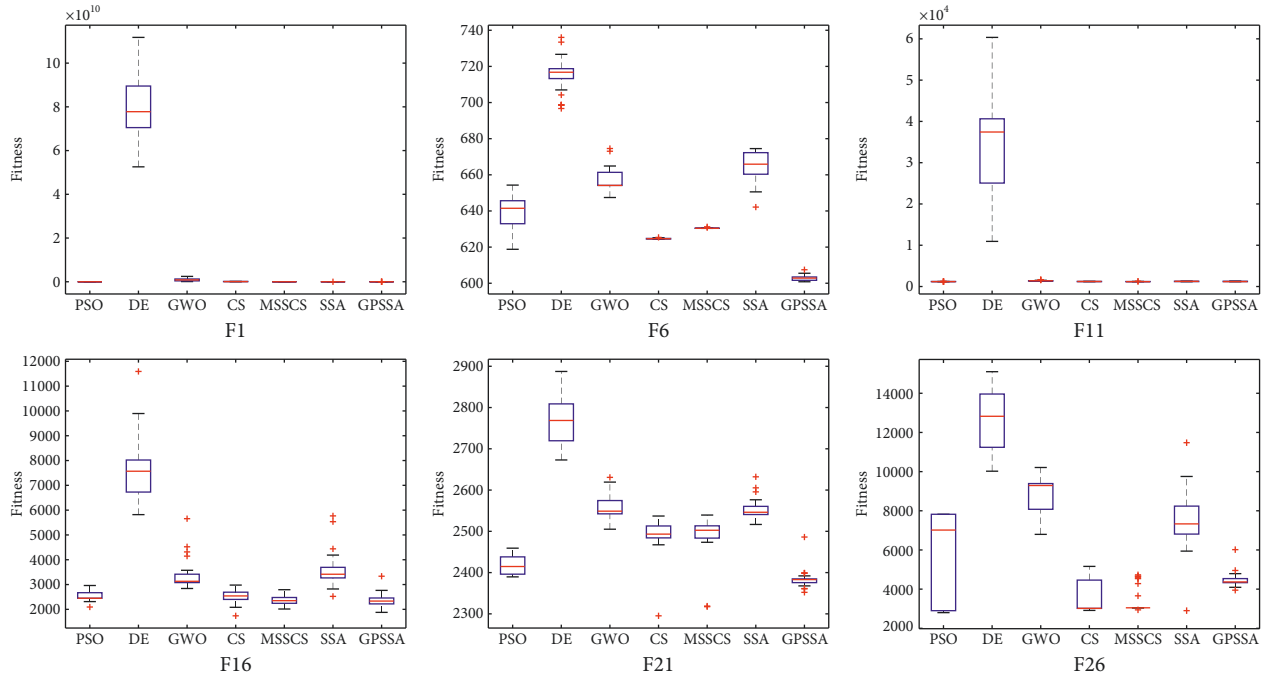


Figure 11: Box diagram of each algorithm.

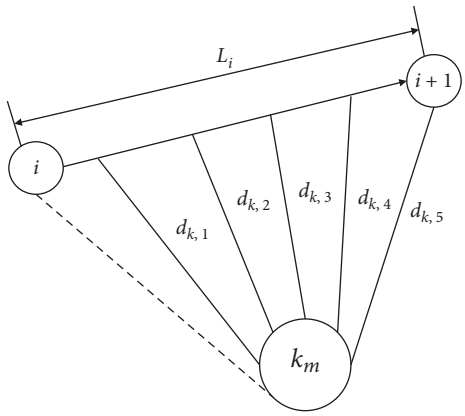


Figure 12: Threat principle.

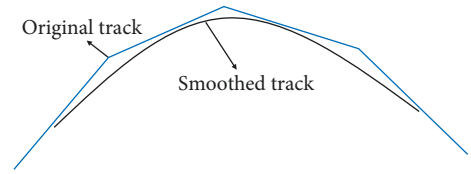


Figure 13: Cubic B-spline principle.

test for GPSSA, and the results are shown in Table 5. When the value is greater than 0.05, it can be considered that there is no significant difference between the two; otherwise, it is considered that there is a significant difference between the two, where "+" and "−" mean greater than and less than 0.05, respectively. Only a few values in the results are greater than 0.05, indicating that GPSSA algorithm is significantly different from other algorithms. Figure 11 shows the selected six groups of box graphs, in which the box graph of GPSSA has a shorter length than other algorithms, indicating that GPSSA algorithm has strong optimization performance and stability. The results show that the optimization performance of GPSSA is better than that of SSA, and the optimization performances of PSO and MSSCS are also better.

In the SSA algorithm, when the participants in the population move closer to the discoverer, they jump directly to the current optimal solution instead of moving slowly to the current optimal solution, so the result of SSA is very poor. This problem leads to a large range of individual update steps of the population in SSA algorithm, which accelerates the convergence of the algorithm, but it is easy to miss the high-quality solution. GPSSA algorithm adopts an improved strategy to make up for this disadvantage, which makes the initial population more uniform, makes full use of other better solutions or generates new solutions, increases the population diversity, makes the search method more flexible, and greatly reduces the loss of population diversity. In general, GPSSA algorithm retains the advantage of fast convergence speed of the original algorithm, has good universality and optimization performance, and has strong competitiveness compared with other algorithms.

7. UAV Path Planning

UAV path planning problem [42] is an optimization problem where UAV uses terrain as cover to effectively avoid various threats, so as to improve the survivability of aircraft and quickly reach the destination. Aiming at the problem of UAV path planning, several methods such as graph theory search, element decomposition, potential field, and natural heuristic algorithm are proposed in the literature [43–45].

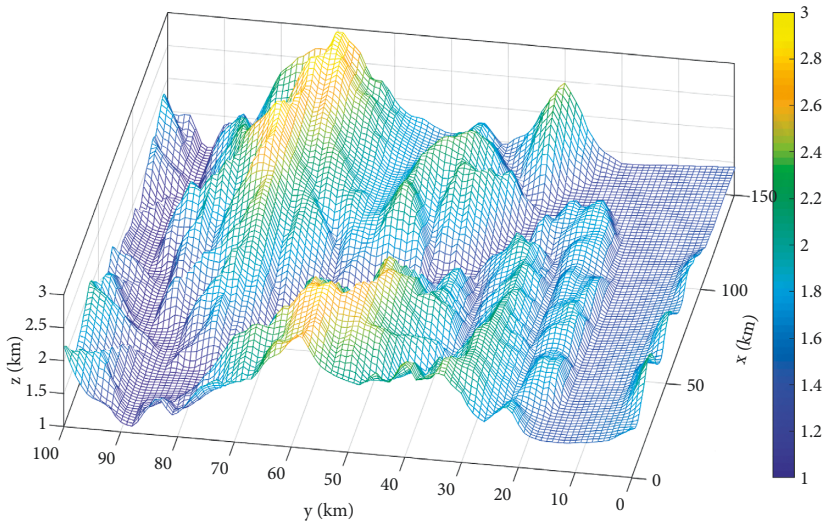


Figure 14: Original map.

Table 6: UAV track planning results.

Index	PSO	DE	GWO	SSA	GPSSA
Best	80.8240	80.6203	80.8124	79.6252	**76.0988**
Worst	105.1887	104.2779	105.1887	107.1757	**89.3498**
Ave	88.6080	87.4127	86.4252	92.5325	**82.4084**
Std	9.1291	8.1642	7.9080	9.3197	**3.6872**

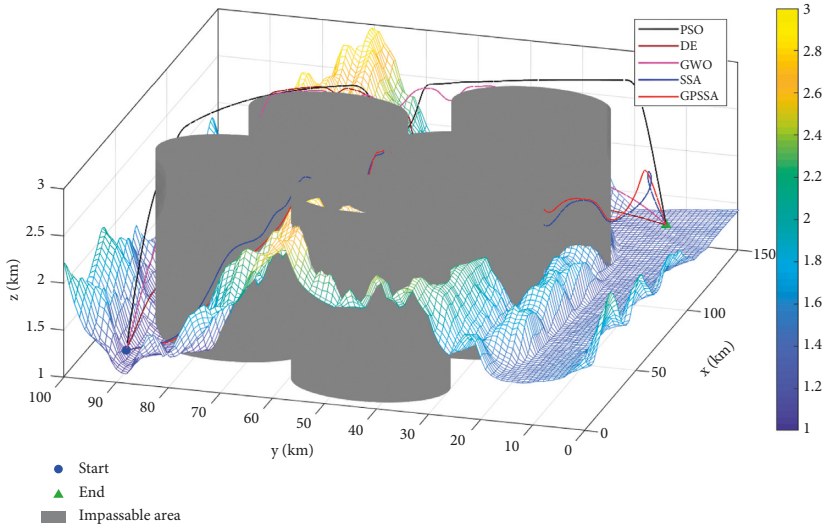


Figure 15: Path planning with threats.

However, with the more and more complex tasks undertaken by UAV, the uncertainty of its flight environment, and the higher requirements for track planning, the superiority of swarm intelligence algorithm in solving complex problems makes more and more scholars apply swarm intelligence algorithm to this field. To solve the UAV track planning problem, it is necessary to establish an appropriate fitness function and consider various constraints affecting the track quality. The static global 3D track planning model mainly includes cost function and constraint function.

7.1. Flight Path Cost. In the actual combat mission, the fuel carried by UAV is limited, the track length can reflect the fuel consumption, and L_i is the track length of the i-th segment. That is, UAV flight fuel consumption cost can be expressed as track length:

$$f_{\text{path}} = \sum_{i=1}^{N} L_i. \tag{17}$$

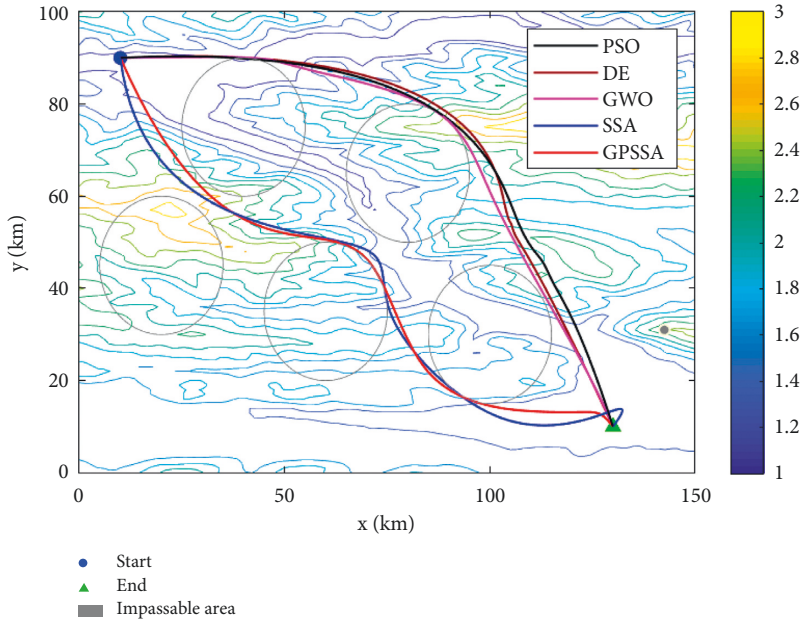


Figure 16: Top view of path planning contour.

7.2. Flight Altitude Change Cost. In order to avoid radar search and prevent collision with mountains or other obstacles, the UAV must raise or lower the height, but repeated lifting and lowering will also endanger the safety of the UAV. The variance of track altitude change can describe the stability of flight altitude. As the cost of altitude change, it can be expressed as

$$f_{\text{height}} = \frac{\sum_{i=1}^{N}\left(z_i - (1/N)\sum_{i=1}^{N} z_i\right)^2}{N}. \tag{18}$$

7.3. Smoothing Cost. In the flight process, the larger the deflection angle is, the more unstable the flight state of the UAV is and the less smooth the flight path is. Therefore, the smoothing cost is added to increase the stability and smoothness of UAV track, and the smoothing cost is expressed by the change degree of deflection angle δ. The function is set as follows:

$$f_{\text{smooth}} = \sum_{i=1}^{N}\left|\delta_i - \delta_{i-1}\right|. \tag{19}$$

7.4. Integrated Threat Constraints. UAV encounters enemy air defense system when it passes through enemy areas, including detection radar, antiaircraft artillery, and ground-to-air missile. The above threats are approximated as a cylindrical area in three-dimensional plane, and the detection or attack range is used as its radius R. The current track segment L_i is divided into five segments, M represents the comprehensive threat, k_M represents the k-th comprehensive threat, R_{k_M} represents the radius of the current threat, and $d_{k,i}$ represents the distance from the current threat point to each of the five equal segments. The

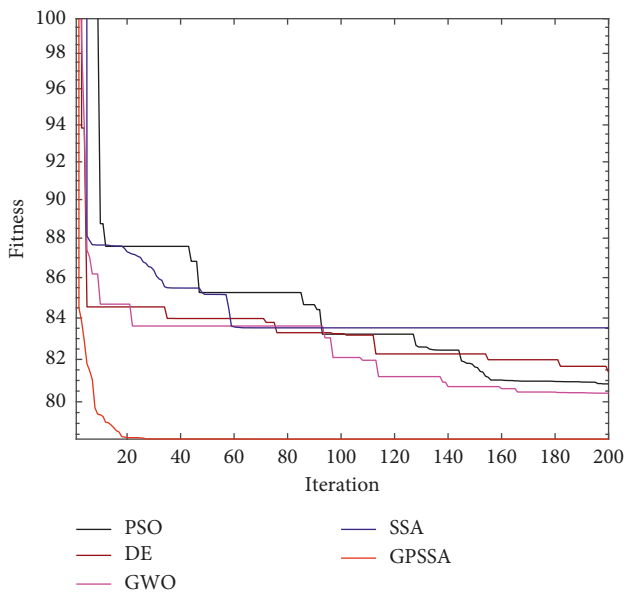


Figure 17: Objective function convergence graph.

threat cost of the current synthetic threat point to the track segment is shown in Figure 12, and its threat constraint function is

$$\text{Constraint} = \sum_{i=1}^{5N}\sum_{k_M=1}^{n_M}\max\left(\frac{R_{k_M} - d_{k_M,i}}{R_{k_M}}, 0\right). \tag{20}$$

η is a penalty function, and the trajectory planning objective function in this paper is

$$\min\left(f_{\text{cost}}\right) = \omega_1 \cdot f_{\text{path}} + \omega_2 \cdot f_{\text{height}} + \omega_3 \cdot f_{\text{smooth}} + \eta \cdot \text{Constraint}. \tag{21}$$

In this paper, the cubic B-spline method is used to ensure better curve smoothness for UAV track smoothing. The spline curve generated during track smoothing is determined by four adjacent control points. By reducing the dependence of track smoothing on each operating point, the smoothed track is more effective, as shown in Figure 13.

7.5. Track Coding. An individual in a population is defined as a path connected by multiple track points, $S_i = [s_{x1}, s_{y1}, s_{z1}, s_{x2}, \ldots\ldots s_{xn}, s_{yn}, s_{zn}]$, in which every 3 coordinates constitute a track point in three dimensions, a total of N track points, and a dimension of $3 * n$ for each individual.

7.6. Experimental Simulation. This paper establishes a three-dimensional mountain terrain using a $100\,\text{km} \times 150\,\text{km} \times 3\,\text{km}$ digital elevation map as shown in Figure 14. There are five nonflight zones for the composite threat, with the center coordinates of (20,45), (40,75), (60,30), (80,65), and (100,30) and a radius of 15. The starting and ending coordinates are (10,90, 1.1) and (130, 10, 1.1).

$\eta = 10^7$, $\omega_1 = 0.5$, $\omega_2 = 0.3$, and $\omega_3 = 0.2$ are set. The cubic B-spline method is used to ensure better curve smoothness for UAV track smoothing.

This paper runs 20 times, with 10 track nodes , 30 dimensions, maximum number of iterations of 200, and a population of 100. The experimental results are shown in Table 6. Path planning with threats is shown in Figure 15. Top view of path planning contour is shown in Figure 16, and target function convergence is shown in Figure 17.

Combining Figures 15–17, it can be seen that SSA is trapped in a local optimum and cannot jump out. GPSSA algorithm can quickly avoid the constraints of threat area and complete path planning, and the path planning is optimal, with faster convergence speed and better accuracy. From the results in Table 6, the four indicators of GPSSA are all optimal, indicating that they have strong optimization ability and stability. The good algorithm performance of GPSSA is verified, which can quickly and accurately get rid of the constraints of threat areas and complete the unmanned aircraft path planning. This helps the unmanned aircraft path static planning to have shorter time, better path planning, and better stability.

8. Conclusions

In this paper, game predatory mechanism and suicide mechanism are proposed to improve sparrow search algorithm, which is more in line with the biological habits of sparrow population, ensures more uniform population through a good point set, makes full use of the better individuals in the population through game predatory mechanism and ensures the flexibility of search, and helps the algorithm get rid of the trap of local optimization through suicide mechanism. An invalid search index is presented for the true location of individuals beyond the boundary during the search process, which improves the defect of the original index and makes it more accurate and realistic to reflect the population diversity, exploration, and development stages in the search process. By comparing the basic test functions and CEC2017 with other heuristic algorithms, it is proved that GPSSA has good optimization performance and effectively improves the shortcomings of poor initial population quality, poor utilization of the current better solution, and poor ability to jump out of local optimal. It also strengthens the ability of the original algorithm to get rid of local optimal attraction while retaining the advantages of fast convergence. Through the UAV path planning simulation, the effectiveness and superiority of GPSSA are verified. Using the advantages of rapid convergence and strong optimization performance of GPSSA, it helps to quickly plan the better path. It provides a new method for the research of this kind of field and provides a good case for the research of this algorithm in other fields.

In the future, we will try to integrate with PSO and other heuristic algorithms and combine the advantages of these two algorithms to improve their poor global search ability, further improve the search performance, and better apply them to more practical problems.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by Regional Foundation of the National Natural Science Foundation of China (no. 61703411).

References

[1] W. L. Liu, Y. J. Gong, W. N. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: A mixed-variable differential evolution approach [J]," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2020.

[2] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times [J]," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1430–1442, 2021.

[3] F. Zhao, R. Ma, and L. Wang, "A self-learning discrete jaya algorithm for multi-objective energy-efficient distributed No-idle flow-shop scheduling problem in heterogeneous factory system [J]," *IEEE Transactions on Cybernetics*, pp. 1–12, 2021.

[4] F. Zhao, X. He, and L. Wang, "A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of No-wait flow-shop problem [J]," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5291–5303, 2021.

[5] F. Zhao, L. Zhao, L. Wang, and H. Song, "An ensemble discrete differential evolution for the distributed blocking

flowshop scheduling with minimizing makespan criterion," *Expert Systems with Applications*, vol. 160, Article ID 113678, 2020.

[6] E. Q. Wu, M. C. Zhou, D. Hu et al., "Self-paced dynamic infinite mixture model for fatigue evaluation of pilots' brains [J]," *IEEE Transactions on Cybernetics*, pp. 1–16, 2021.

[7] X. Zhang, H. Wang, C. Du et al., "Custom-molded offloading footwear effectively prevents recurrence and amputation, and lowers mortality rates in high-risk diabetic foot patients: A multicenter, prospective observational study," *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy*, vol. 15, pp. 103–109, 2022.

[8] X. Ran, X. Zhou, M. Lei, W. Tepsan, and W. Deng, "A novel K-means clustering algorithm with a noise algorithm for capturing urban hotspots," *Applied Sciences*, vol. 11, no. 23, p. 11202, 2021.

[9] W. Deng, X. Zhang, Y. Zhou et al., "An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems," *Information Sciences*, vol. 585, pp. 441–453, 2022.

[10] R. Salgotra and U. Singh, "The naked mole-rat algorithm," *Neural Computing & Applications*, vol. 31, no. 12, pp. 8837–8857, 2019.

[11] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the International symposium on stochastic algorithms*, pp. 169–178, Springer, Sapporo, Japan, 26 October 2009.

[12] S. Mirjalili, "The Ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[14] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[15] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, 2016.

[16] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[17] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[18] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: V," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.

[19] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, Article ID 115079, 2021.

[20] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The Colony predation algorithm," *Journal of Bionics Engineering*, vol. 18, no. 3, pp. 674–710, 2021.

[21] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[22] X. S. Yang and X. He, "Bat algorithm: Literature review and applications," *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, pp. 141–149, 2013.

[23] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[24] Y. Li, S. Wang, Q. Chen et al., "Comparative study on some new swarm intelligence optimization algorithms [J]," *Computer engineering and application*, vol. 56, no. 22, pp. 1–12, 2020.

[25] Z. Wang, X. Huang, and D. Zhu, "A multistrategy-integrated learning sparrow search algorithm and optimization of engineering problems [J]," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–21, 2022.

[26] X. Lv, X. Mu, and J. Zhang, "Multi-threshold image segmentation based on improved sparrow search algorithm [J]," *Systems Engineering and Electronics*, vol. 43, no. 2, pp. 318–327, 2021.

[27] S. Yan, P. Yang, D. Zhu, W. Zheng, and F. Wu, "Improved sparrow search algorithm based on iterative local search [J]," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–31, 2021.

[28] C. Ouyang, D. Zhu, and F. Wang, "A learning sparrow search algorithm [J]," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–23, 2021.

[29] G. Chen, D. Lin, F. Chen, and X. Chen, "Image segmentation based on logistic regression sparrow algorithm [J]," *Journal of Beijing University of Aeronautics and Astronautics*, pp. 1–14, 2021.

[30] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization [J]," *National University of Defense Technology*, Technical Report, 2017.

[31] N. H. Awad, M. Z. Ali, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Technical Report, 2016.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization [C],"vol. 4, pp. 1942–1948, in *Proceedings of the ICNN'95-international conference on neural networks*, vol. 4, IEEE, Perth, WA, Australia, November - December 1995.

[33] K. V. Price, "Differential evolution," *Handbook of Optimization*, Springer, Berlin, Heidelberg, pp. 187–214, 2013.

[34] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[35] H. Peng, Z. Zeng, C. Deng, and Z. Wu, "Multi-strategy serial cuckoo search algorithm for global optimization," *Knowledge-Based Systems*, vol. 214, Article ID 106729, 2021.

[36] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization [J]," *Applied Soft Computing*, vol. 107, no. 2, Article ID 107376, 2021.

[37] L. Hua and Y. Wang, *Application of Number Theory in Modern Analysis [M]*, pp. 1–99, Science Press, Beijing, China, 1978.

[38] R. Greenberg, "Feeding neophobia and ecological plasticity: A test of the hypothesis with captive sparrows," *Animal Behaviour*, vol. 39, no. 2, pp. 375–379, 1990.

[39] S. Cheng, Y. Shi, Q. Qin, Q. Zhang, and R. Bai, "Population diversity maintenance in brain storm optimization algorithm," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 4, no. 2, pp. 83–97, 2014.

[40] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "On the exploration and exploitation in popular swarm-based metaheuristic algorithms," *Neural Computing & Applications*, vol. 31, no. 11, pp. 7665–7683, 2019.

[41] V. Stanovov, S. Akhmedova, and E. Semenkin, "LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems [C]," in *Proceedings of*

the 2018 IEEE congress on evolutionary computation (CEC), pp. 1–8, IEEE, Rio de Janeiro, Brazil, 10 July 2018.

[42] U. Cekmez, M. Ozsiginan, and O. K. Sahingoz, "A UAV path planning with parallel ACO algorithm on CUDA platform [C]," in *Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 347–354, IEEE, Orlando, FL, USA, 27 May 2014.

[43] R. W. Beard, T. W. McLain, M. A. Goodrich, and E. P. Anderson, "Coordinated target assignment and intercept for unmanned air vehicles [J]," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 911–922, 2002.

[44] T. W. McLain and R. W. Beard, "Coordination variables, coordination functions, and cooperative timing missions," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 150–161, 2005.

[45] Y. Lin and S. Saripalli, "Sampling-based path planning for UAV collision avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3179–3192, 2017.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2286818, 9 pages
https://doi.org/10.1155/2022/2286818

Research Article

PyraPVConv: Efficient 3D Point Cloud Perception with Pyramid Voxel Convolution and Sharable Attention

Yuhong Chen,[1] Weilong Peng,[1] Keke Tang,[1] Asad Khan,[1] Guodong Wei,[2] and Meie Fang[1]

[1]*Guangzhou University, Guangzhou, China*
[2]*South China University of Technology, Guangzhou, China*

Correspondence should be addressed to Keke Tang; tangbohutbh@gmail.com and Meie Fang; fme@gzhu.edu.cn

Yuhong Chen and Weilong Peng contributed equally to this work.

Received 5 January 2022; Revised 4 February 2022; Accepted 7 February 2022; Published 13 May 2022

Academic Editor: Diego Oliva

Designing efficient deep learning models for 3D point cloud perception is becoming a major research direction. Point-voxel convolution (PVConv) Liu et al. (2019) is a pioneering research work in this topic. However, since with quite a few layers of simple 3D convolutions and linear point-voxel feature fusion operations, it still has considerable room for improvement in performance. In this paper, we propose a novel pyramid point-voxel convolution (PyraPVConv) block with two key structural modifications to address the above issues. First, PyraPVConv uses a voxel pyramid module to fully extract voxel features in the manner of feature pyramid, such that sufficient voxel features can be obtained efficiently. Second, a sharable attention module is utilized to capture compatible features between multi-scale voxels in pyramid and point cloud for aggregation, as well as to reduce the complexity via structure sharing. Extensive results on three point cloud perception tasks, i.e., indoor scene segmentation, object part segmentation and 3D object detection, validate that the networks constructed by stacking PyraPVConv blocks are efficient in terms of both GPU memory consumption and computational complexity, and are superior to the state-of-the-art methods.

1. Introduction

With the advance of depth sensing devices, 3D point clouds can be captured in a much easier manner. Therefore, applications of 3D point cloud perception are now booming, e.g., simultaneous localization and mapping (SLAM) [1–3], and autonomous driving [4–6]. In the last few decades, 3D point cloud perception mainly depends on hand-crafted shape descriptors [7–9]. Until very recently, researchers start to extend deep learning models which are mature in the field of 2D computer vision to handle 3D perception tasks, significantly refreshing the state-of-the-art records [10–12]. However, 3D deep learning models with higher accuracies always have higher complexities, which now becomes the major obstacle for their application to real-world scenarios.

There have been many attempts that utilize deep networks to perceive 3D point clouds. According to the representation of data that feeds to deep networks, these methods could be divided into two categories broadly: structure-based methods and point-based methods. Structure-based methods first convert irregular point clouds into structured grid representations, e.g., projecting point clouds into bird's eye views [13, 14] or rasterizing into 3D voxel grids [10, 15–17], and then adopt traditional 2D convolutional neural networks (CNNs) or their simple extensions to extract discriminative CNN features. However, it introduces exponential computational cost and memory for detailed 3D geometric learning at high resolutions. Point-based methods instead impose multi-layer perceptrons (MLPs) followed with maximum pooling [11, 18] or irregular kernels [19] to 3D point clouds, such that they can handle points directly. Nevertheless, since these methods require accessing irregularly scattered points especially for local feature aggregation, they are also inefficient.

By analysing both the advantages and disadvantages of structure-based and point-based methods, the pioneering point-voxel convolution (PVConv) [20] proposes to combine them together for efficiency purpose. Particularly, point-based MLPs are adopted to extract 3D features in the point branch; and voxel convolutions aggregate local features coarsely in the voxel branch; in addition, linear interpolation is conducted to fuse the features of the two branches. PVConv brings large improvements in terms of GPU memory consumption and computational efficiency, but the accuracy of PVConv is somewhat sacrificed. By going through the structure of PVConv, we hypothesize that there are two key factors that limit its performance. First, the voxel-based networks with strong feature extraction capabilities are utilized too conservatively, i.e., with only two layers of simple 3D convolutions. Second, the voxel features that are already insufficient would further lose during the fusion process, since it is implemented via a linear-based interpolation that is not powerful enough.

To resolve the above two issues, we intentionally design a novel pyramid point-voxel convolution (PyraPVConv) block. First, we adopt a more powerful 3D voxel convolution branch that extracts multi-scale voxel features in the form of feature pyramid, such that powerful 3D voxel convolutions can be fully exploited with sacrificing moderate additional computational overhead. Second, to alleviate the information loss during the feature fusion process between point and voxel branches, we propose utilizing the attention mechanism to learn to combine them in a more compatible way. Particularly, we design a sharable attention module that learns the relevant scores between multiple voxel branches and the point branch with sharing a structure, such that it reduces the overhead required by multibranch attention. With these two designs against the weaknesses of PVConv, the PyraPVConv block can perceive 3D point clouds in a better tradeoff between accuracy and efficiency.

Overall, our contribution is three-fold.

(i) We propose a lightweight 3D perception block, PyraPVConv, that can perform 3D point cloud perception accurately and efficiently

(ii) We design a voxel pyramid module, which better extracts voxel features without introducing too much additional computational overhead

(iii) We devise a sharable attention module that fuses the features of point branch and multiple voxel branches in a more effective nonlinear manner

We construct PyraPVCNN by stacking multiple PyraPVConv blocks following PVCNN [20], and evaluate it on various point cloud perception tasks, e.g., indoor scene and object part segmentation, 3D object detection. Extensive experiments demonstrate the superiority of PyraPVConv to the state-of-the-art methods in terms of both accuracy and efficiency.

2. Related Work

2.1. Efficient Deep Learning for Point Clouds. Deep learning techniques have been widely adopted for handling 3D point cloud perception tasks, e.g., classification and segmentation, by projecting sparse point clouds into compact semantic representations [21–25]. However, as indicated in [20], most current 3D deep learning methods for point clouds are less efficient, e.g., voxel-based methods require large amounts of memory for maintaining detailed 3D structures, and point-based methods require high CPU latency to search neighborhood points for feature aggregation.

To perceive 3D point clouds in a more efficient way, Liu et al. [20] proposed the point-voxel convolution (PVConv), that combines point-based MLPs for individual point feature extraction and voxel-based CNNs for neighborhood feature aggregation. Since with only two layers of simple 3D CNNs, the performance of PVConv is somewhat sacrificed. To better utilize the voxel information, Tang et al. [26] further designed sparse point-voxel convolution (SPVConv), which uses sparse convolution to handle high-resolution voxels on the basis of PVConv, and further adopted network architecture search (NAS) for searching the best architecture. Although their method showed good performance in large-scale scenarios, sparse convolution is complex and is less efficient in common scenes. Our method also aims to fully utilize 3D voxel information. Differently, we adopt the concept of feature pyramid to balance the accuracy and efficiency.

We also notice some other works that also aim to efficient point cloud learning by randomly key point sampling [27], by irregularly volume partition [28], and by leveraging mature 2D methods [29], etc., but these directions are out of our scope.

2.2. Multiscale Feature Modeling. Modeling multi-scale features has been validated to be a useful strategy in computer vision, e.g., maintaining scale-invariant property in SIFT [30] and controlling the receptive field in CNNs [31]. This strategy is also widely adopted in 3D deep learning. PointNet++ [18] extracts multi-scale features of point clouds by hierarchically applying PointNets [11]. 3D object detection frameworks [17, 32, 33] adopt different detection heads with multi-scale feature maps to handle both large and small object classes. ContFuse [34] uses continuous convolution to aggregate multi-scale feature maps from different ResNet blocks [35]. By extending RPN-FPN module [36] to 3D, Voxel-FPN [37] uses feature pyramid to aggregate voxel features of different voxelization resolutions. We also utilize the multi-scale feature modeling strategy, but are to balance the accuracy and efficiency of the voxel branch for point-voxel convolution, which has not been investigated before.

2.3. Attention Mechanism. Attention is originally a physiological mechanism that describes the phenomenon that humans' perception system could focus on the object of

interest while suppressing the background [38]. Inspired by it, deep learning researchers attempt to exploit it to analyze networks' focus [39, 40] or enforce neural networks to focus on more important features [41, 42].

Attention is also widely adopted to model the relationship between two instances, e.g., generating the most relevant sentences for images in the task of image caption [43] or searching the most relevant sentences between two different languages in the task of machine translation [44]. By projecting the query instance into the same high-dimensional space as the target, relevant spatial regions of the query will be highlighted to guide the desired inference. We also aim to model the relationship. Differently, our inferred relevance is used for the fusion of features extracted from the voxel branch and point branch during point-voxel convolution.

3. Method

In this section, we will first review the architecture of PVConv, analyze the factors that may limit its performance and then introduce the overview of our solution, i.e., pyramid point-voxel convolution (PyraPVConv). After that, we describe the two main components of PyraPVConv: the voxel pyramid module and the sharable attention module.

3.1. Review of PVConv [20]. Methodology of PVConv Given an unordered point set $\mathbf{P} = \{(p_k, f_k)\}$ with $\{p_k\}$ denoting the point coordinates and $\{f_k\}$ as the point features, PVConv adopts an isolated *point branch* to extract individual point features using MLPs similar to PointNet. Apart from that, another *voxel branch* in PVConv is utilized to facilitate efficient and powerful neighbourhood feature aggregation. Particularly, the feeded voxels to the voxel branch are generated by first normalizing the point coordinates and then conducting voxelization to transform the normalized point cloud $\{(\widehat{p}_k, f_k)\}$ into a volume $\mathbf{V}$ by averaging all features $\{f_k\}$ whose normalized coordinates $\{\widehat{p}_k\}$ fall into that voxel grid.

Finally, the two-branch features are linearly fused, i.e., the voxel features are devoxelized to the point cloud domain using trilinear interpolation, and then added with point features.

Discussion on PVConv As mentioned in PVConv, the MLPs in the *point branch* can already output discriminative features for each point. Therefore, the main contribution of PVConv is to leverage *the voxel branch* for neighborhood feature aggregation.

However, perhaps focusing too much on the efficiency factor, PVConv is *conservative* in utilizing the *voxel branch*, e.g., very limited volume resolutions and 3D convolution layers are adopted. Indeed, 3D convolutions that are derived from mature 2D convolution techniques are powerful, and have a good tradeoff between accuracy and efficiency. Furthermore, simple trilinear interpolation would lead to information loss of those voxel features that are not sufficient originally.

3.2. Overview of PyraPVConv. To resolve the above issues of PVConv, we propose a novel PyraPVConv block with two key structural modifications: a voxel pyramid module for the voxel branch and a sharable attention module for feature fusion. Similar as in PVConv, the point branch of PyraPVConv extracts individual point features. At the same time, the voxel branch of PyraPVConv adopts the voxel pyramid module to extract more powerful voxel features in pyramid. Then, the shareable attention module fuses point features and voxel features in pyramid nonlinearly. Please refer to Figure 1 for a demonstration.

3.3. Voxel Pyramid Module. To extract sufficient voxel features without bringing too much additional computational overhead, we propose a voxel pyramid module to capture multiscale features in different pyramid levels, inspired by [36]. Please refer to Figure 2 for a demonstration.

Volume Generation Given the normalized point cloud $\{(\widehat{p}_k, f_k)\}$ as in PVConv, we generate a volume by averaging all features $\{f_k\}$ whose coordinates $\{\widehat{p}_k\}$ fall into that voxel grid via the following equation:

$$V_r(u,v,w) = \frac{1}{N_{r,u,v,w}} \sum_{k=1}^{N} \mathbb{I}\left(\lfloor x_k \times r \rfloor = u, \lfloor y_k \times r \rfloor = v, \lfloor z_k \times r \rfloor = w\right) \times f_{k,c}, \tag{1}$$

where r denotes the volume resolution, $N_{r,u,v,w}$ denotes the number of points falling in the voxel grid (u, v, w) of volume $\mathbf{V}_r$, $f_{k,c}$ denotes the c^{th} channel corresponding to $\widehat{p}_k$, and $\mathbb{I}$ is a binary indicator function.

Bottom-up Feature Extraction Given $\mathbf{V}_r$, we feed it to the same 3D voxel convolution networks as in PVConv, i.e., two groups of conv3d, batch normalization and activation layers, to obtain 3D voxel feature $f_r^{\mathbf{V}}$,

$$f_r^{\mathbf{V}} = \text{Conv3D}_1(\mathbf{V}_r). \tag{2}$$

Unlike PVConv that adopts a conservative strategy in the voxel branch, we suggest making full use of the features in volume $\mathbf{V}_r$. Considering the efficiency, we conduct an average pooling operation with a scaling rate of 1/2 to $f_r^{\mathbf{V}}$, and then feed it another 3D voxel convolution networks,

$$f_{r/2}^{\mathbf{V}} = \text{Conv3D}_2\left(\text{MaxPool}\left(f\begin{pmatrix}\mathbf{V}\\ r\end{pmatrix}\right)\right). \tag{3}$$

We apply the same operations iteratively to obtain multiscale 3D voxel features. In this paper, we adopt three-scale features $f_r^{\mathbf{V}}$, $f_{r/2}^{\mathbf{V}}$, and $f_{r/4}^{\mathbf{V}}$.

To-down Feature Aggregation Given each 3D voxel feature, we enhance it by aggregating with the feature of a higher pyramid level (if has) that is spatially coarser but semantically strong. It is implemented by applying an upsampling operation followed with addition,

$$\begin{aligned} \widehat{f}_{r/4}^{\mathbf{V}} &= f_{r/4}^{\mathbf{V}}, \\ \widehat{f}_{r/2}^{\mathbf{V}} &= \text{UpPool}\left(\widehat{f}_{r/4}^{\mathbf{V}}\right) + f_{r/2}^{\mathbf{V}}, \\ \widehat{f}_{r}^{\mathbf{V}} &= \text{UpPool}\left(\widehat{f}_{r/2}^{\mathbf{V}}\right) + f_{r}^{\mathbf{V}}. \end{aligned} \tag{4}$$

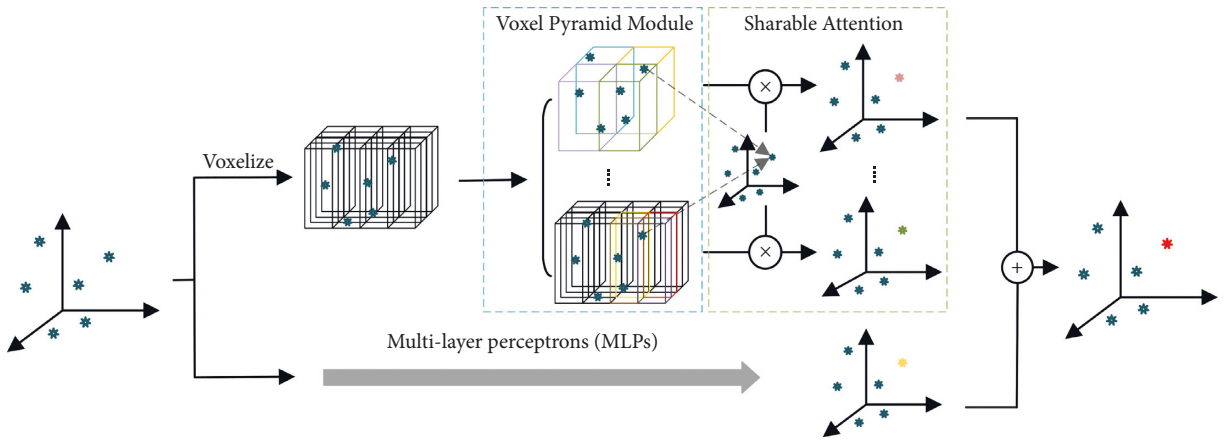


Figure 1: The architecture of PyraPVConv, with a point branch extracting single point features, and a voxel branch aggregating neighborhood information via a voxel pyramid module and an sharable attention mechanism.

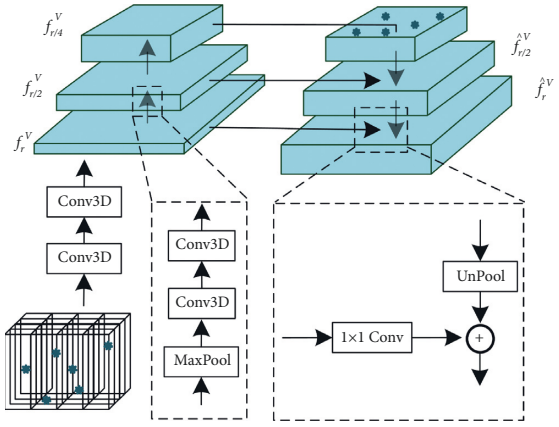


Figure 2: Demonstration of the bottom-up feature extraction and top-down feature aggregation in the voxel pyramid module.

Therefore, we obtain three semantic enhanced voxel features: $\widehat{f}_r^{\mathbf{V}}$, $\widehat{f}_{r/2}^{\mathbf{V}}$, and $\widehat{f}_{r/4}^{\mathbf{V}}$.

3.4. Sharable Attention Module. With enhanced voxel features $\widehat{f}_r^{\mathbf{V}}$, $\widehat{f}_{r/2}^{\mathbf{V}}$, $\widehat{f}_{r/4}^{\mathbf{V}}$ and point features $\{f_k\}$ as inputs, we fuse them via using the attention mechanism. Since voxel features in different pyramid levels contain discriminative but complementary information, we apply attentive fusion of them with point features separately. It is implemented by converting the voxel features into the same feature space as point features, and then summarize them. See Figure 3 for a demonstration. In the following, we will describe the attention method taking $\widehat{f}_r^{\mathbf{V}}$ as an example.

Attentive Fusion for a Pyramid Level Instead of fusing all voxel features, we only consider eight neighboring voxel features $\left\{\widehat{f}_r^{\mathbf{V}}(k,m)\right\}$ with $m = 1, 2, \ldots, 8$ for each point $\{(\widehat{p}_k, f_k)\}$ following PVConv, considering the tradeoff between efficiency and performance. Specifically, we adopt a similar attention process to that in [45] with three key steps. First, we project point feature f_k and its eight neighboring voxel features $\left\{\widehat{f}_r^{\mathbf{V}}(k,m)\right\}$ into the same feature space using two networks Q_r and K_r. Second, we calculate the relevant scores between eight voxel features and that point feature by conducting dot product of their flattened features. Note that, we also adopt the Sigmoid operation to enforce the relevant scores to be between 0 and 1.

$$\mathrm{Rel}\left(f_k, \widehat{f}_r^{\mathbf{V}}(k,m)\right) = \mathrm{Sigmoid}\left(Q_r(f_k) \times K_r\left(\widehat{f}_r^{\mathbf{V}}(k,m)\right)\right). \quad (5)$$

Finally, we convert the voxel features $\left\{\widehat{f}_r^{\mathbf{V}}(k,m)\right\}$ into a summable space with point features using the network H_r, and then aggregate them weighted with the relevant scores $\mathrm{Rel}(f_k, \widehat{f}_r^{\mathbf{V}}(k,m))$.

$$f_k' = f_k + \sum_m H_r\left(\widehat{f}_r^{\mathbf{V}}(k,m)\right)\mathrm{Rel}\left(f_k, \widehat{f}_r^{\mathbf{V}}(k,m)\right). \quad (6)$$

Multiple Pyramid Fusion with Sharable Attention For voxel features in all three pyramid levels, we apply the above attentive fusion method, and then summarize them together.

$$\widehat{f}_k' = f_k + \sum_r \sum_m H_r\left(\widehat{f}_r^{\mathbf{V}}(k,m)\right)\mathrm{Rel}\left(f_k, \widehat{f}_r^{\mathbf{V}}(k,m)\right). \quad (7)$$

Therefore, $\widehat{f}_k'$ is the final fused feature that contains both point and voxel features. Particularly, since the inputs for networks Q_t with $t = r, r/2, r/4$ are exactly the same, we simply use one network Q and share it with all three pyramid levels for efficiency purpose.

4. Experimental Results

To validate the effectiveness and efficiency of PyraPVConv, we extensively evaluate the performance of PyraPVCNN, which is constructed by stacking multiple PyraPVConv blocks, on three different tasks, i.e., indoor scene segmentation, object part segmentation and 3D object detection.

4.1. Implementations. For PyraPVCNN, we replace all the PVConv blocks in PVCNN with PyraPVConv and use the same decoding layer. We implement PyraPVCNN and reproduce all the evaluated networks with PyTorch for fair comparisons, and report the latency and memory

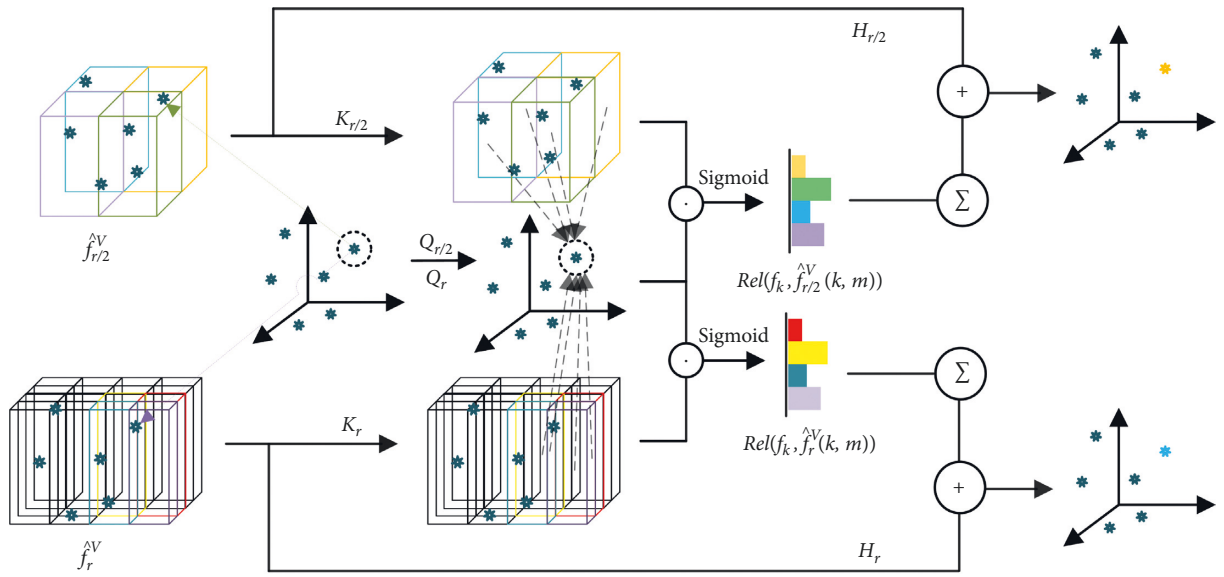


Figure 3: Demonstration of the sharable attention module.

consumption at test time on a workstation with an Intel Xeon E5-2678 CPU@2.5 Hz and 64 GB of memory using a single RTX 2080Ti GPU.

4.2. Indoor Scene Segmentation. Setups We conduct semantic segmentation experiments on the large-scale Stanford 3D semantic parsing dataset (S3DIS) [47, 48]. S3DIS is scanned across 271 rooms in 6 areas with a Matterport camera (combined with 3 structured light sensors at different intervals), and then each point in the scanned point cloud is annotated with a semantic label (e.g., a total of 13 objects such as chairs, tables, floors, walls, etc.). We adopt the same data processing procedure as PVCNN [20]. We train the models on regions 1, 2, 3, 4, and 6 and test them on region 5 to evaluate the mean intersection-over-union (mIoU) and the mean class-wise accuracy (mAcc), with both metrics measured as percentages.

Models Six state-of-the-art methods are adopted as baselines: PointNet [20], PointNet++ [18], DGCNN [25], PointCNN [46], PVCNN [20] and RandLA-Net [27]. Since our focus is on the efficiency of deep networks, similar as in PVCNN, we also evaluate the narrower versions of PVCNN and PyraPVCNN by reducing the number of feature channels from that of the original version (denoted as $1 \times C$) to 12.5% (denoted as $0.125 \times C$), 25% (denoted as $0.25 \times C$) and 50% (denoted as $0.5 \times C$).Comparison Results The results reported in Table 1 show that Ours $(0.125 \times C)$ performs better than two state-of-the-art point-based methods: PointNet [11] and DGCNN [25] and voxel-based methods: PVCNN $(0.25 \times C)$ [20], and is comparable to the state-of-the-art RandLA-Net [27], but with less latency and GPU memory consumption. In addition, the performance of Ours $(1 \times C)$ is nearly 10% higher than that of DGCNN [25], but the latency is reduced by 20%, and GPU memory consumption is reduced by 1.6 times. In particular, Ours $(0.25 \times C)$ outperforms the full version of PVCNN, but the latency and GPU memory consumption are much less. Please refer to Figure 4 to see an illustrative demonstration of the tradeoff between accuracy (mIoU) and the incurred overhead (the number of parameters, latency and GPU memory consumption).

We also visualize the segmentation results in Figure 5. It could be seen that Ours $(0.25 \times C)$ can better utilize neighborhood information to improve the prediction of point labels, compared with PVCNN.

Ablation Studies To validate the importance of the two key modules of PyraPVConv: voxel pyramid module and the sharable attention module, we report the results of ablation studies using two narrow versions of the PyraPVCNN (i.e., $0.25 \times C$ and $0.5 \times C$) in Table 2. It could be seen that the performance will drop, if we delete any one of them. In particular, the voxel pyramid module has a slightly larger impact on the performance.

4.3. Object Part Segmentation. Setups We choose the ShapeNet Parts dataset [49] to conduct the experiment of object part segmentation. ShapeNet Parts is a collection of 16681 point clouds selected from 16 categories of the ShapeNetCore, and is manually annotated with a total of 50 parts. Following [46], we train the models on 14006 point clouds, evaluate the part-averaged IoU for each of the remaining 2874 point clouds, and then average them as the final metric, i.e., mIoU (%).

Models We choose point-based models: PointNet [11], PointNet++ [18], and DGCNN [25] and the voxel-based model: 3D-UNet [50], and PVCNN [20] as baselines.

Comparison Results As shown in Table 3, Ours $(0.25 \times C)$ performs much better while the latency is 36 times lower and the GPU memory consumption is 11.7% lower than that of 3D-UNet. Moreover, it is superior to the most advanced point-based methods such as PointNet [11], PointNet++ [18] and DGCNN [25] in all aspects. In particular, we would like to emphasize that Ours $(0.5 \times C)$ outperforms PVCNN [20], and

Table 1: Indoor scene segmentation results on the S3DIS dataset. Note that the input data of PointCNN [46] include 16×2048 points, while the data of the other methods include 8×4096 points.

	mAcc	mIoU	Latency (ms)	GPU mem. (GB)	#Param.
PointNet	82.54	42.97	16.1	1.50	3.53 M
PVCNN $(0.125 \times C)$	82.60	46.94	7.6	0.46	43.16 K
DGCNN	83.64	47.94	79.0	4.50	987.00 K
PVCNN $(0.25 \times C)$	84.52	51.96	11.4	0.76	166.21 K
Ours $(0.125 \times C)$	84.88	**52.16**	10.3	0.59	143.19 K
PVCNN $(0.5 \times C)$	85.88	54.73	17.5	0.97	650.35 K
PVCNN	86.25	55.54	35.9	1.92	2.57 M
Ours $(0.25 \times C)$	86.49	**55.86**	21.6	1.04	693.72 K
Ours$(0.5 \times C)$	86.93	57.02	41.3	1.84	1.82 M
PointCNN	85.91	57.26	282.3	4.60	5.86 M
RandLA-Net	85.10	**58.60**	911.1	2.57	4.76 M
Ours $(1 \times C)$	**86.96**	57.98	71.2	2.52	3.13 M

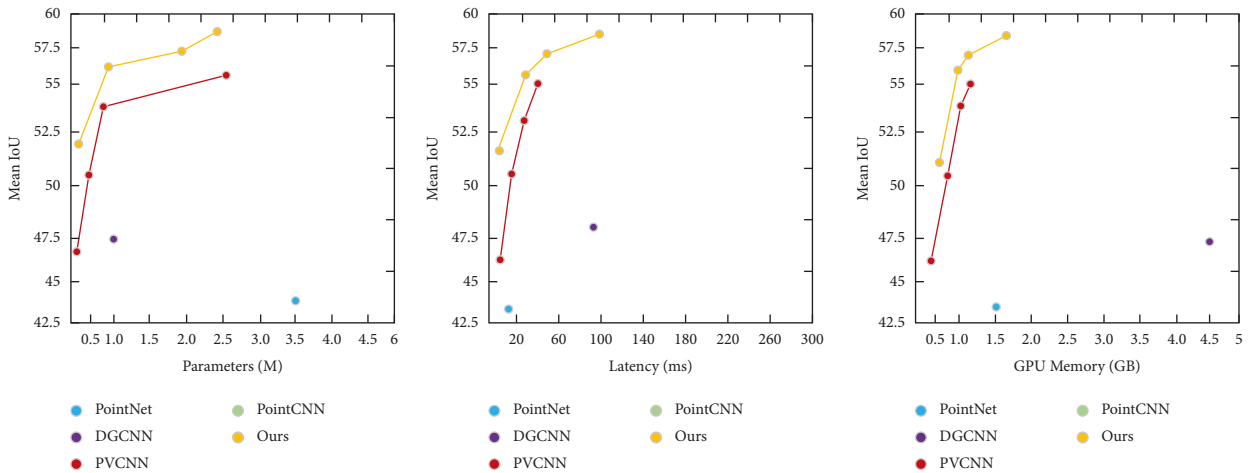


Figure 4: The tradeoff between accuracy and the number of parameters, measured latency, and GPU memory consumption for PyraPVCNN and the state-of-the-art baselines on S3DIS. (a) Parameters (M), (b) Latency (ms), (c) GPU memory (GB).

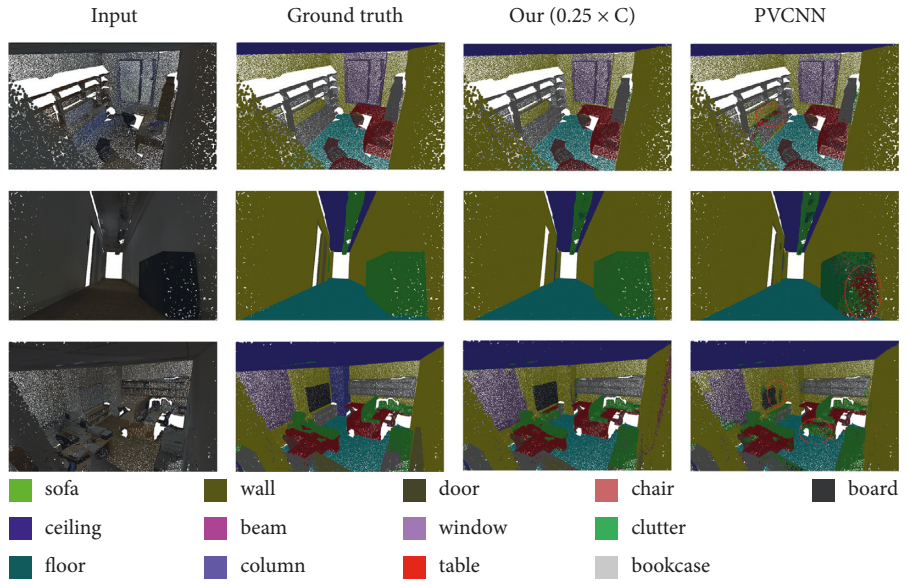


Figure 5: Visualization of the semantic segmentation results on the S3DIS dataset.

Table 2: The indoor scene segmentation performance on the S3DIS dataset with ablating the voxel pyramid module (VPM.) and the sharable attention module (ASM.).

Channel	Metric	Ours	W/o VPM.	W/o ASM.
$0.25 \times C$	mIoU	**55.86**	53.01	53.87
	mAcc	**86.49**	84.91	85.56
$0.5 \times C$	mIoU	**57.02**	55.07	55.68
	mAcc	**86.93**	85.67	86.31

Table 3: Object part segmentation results on the ShapeNet Parts dataset.

	Input data	Convolution	mIoU	Latency (ms)	GPU mem. (GB)
PointNet	Points (8×4096)	None	83.70	16.0	1.5
3D-UNet	Voxels (8×963)	Voxel-based	84.60	480.0	17.6
PointNet++	Points (8×4096)	Point-based	84.70	55.6	4.0
DGCNN	Points (8×4096)	Point-based	84.70	61.8	4.8
PVCNN ($0.25 \times C$)	Points (8×4096)	Voxel-based	84.72	10.1	1.2
Ours($0.25 \times C$)	Points (8×4096)	Voxel-based	**85.12**	13.2	1.5
PVCNN ($0.5 \times C$)	Points (8×4096)	Voxel-based	85.38	17.5	1.8
PVCNN	Points (8×4096)	Voxel-based	85.70	35.2	3.1
Ours ($0.5 \times C$)	Points (8×4096)	Voxel-based	**85.97**	29.7	2.4
PointCNN	Points (16×2048)	Point-based	86.10	95.6	5.1
Ours ($1 \times C$)	Points (8×4096)	Voxel-based	**86.82**	56.7	3.2

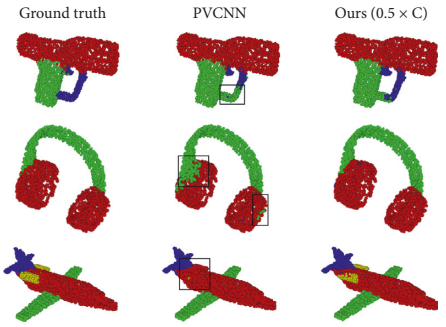


Figure 6: Visualization of object part segmentation results on the ShapeNet Parts dataset.

Table 4: 3D object detection results of the Frustum Networks [29] on the validation set of KITTI with different backbones.

Backbone	Efficiency		Car			Pedestrian			Cyclist		
	Latency (ms)	GPU mem. (GB)	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointNet	29.1	1.5	83.26	69.28	62.56	65.08	55.85	49.28	74.54	55.95	52.65
PointNet++	101.2	2.5	83.76	70.92	63.65	70.00	61.32	53.59	77.15	56.49	53.37
PVCNN	51.9	1.9	84.22	71.11	**63.66**	69.16	60.28	52.52	78.67	57.79	54.16
Ours ($0.25 \times C$)	**39.6**	**1.4**	**85.42**	**71.24**	63.09	**71.80**	**64.67**	**56.87**	**79.32**	**58.57**	**55.01**

Ours ($1 \times C$) outperforms PointCNN [46] with only about 58% latency and GPU memory consumption. Please refer to the visualization results in Figure 6 for intuitive comparisons.

4.4. 3D Object Detection. Setups We conduct 3D object detection experiments on the large outdoor dataset KITTI [51]. KITTI provides 7481 training and verification samples for objects such as cars, pedestrians, and cyclists, of which 3711 samples are used for training and the remaining 3769 samples are for verification. We evaluate all models 20 times and report the average 3D average accuracy (AP) measured as percentages.

Models We build multiple Frustum Networks [29] with PointNet [11], PointNet++ [18], PVCNN [20], a narrow version of PyraPVCNN ($0.25 \times C$) as backbones.

Comparison Results As shown in Table 4, the Frustum Network with PyraPVCNN ($0.25 \times C$) (denoted as F-PyraPVCNN) as backbone outperforms than all three other Frustum Networks with PointNet [11], PointNet++ [18] and PVCNN [20] as backbones (denoted as F-PointNet, F-PointNet++ and F-PVCNN), but with much less latency and GPU memory consumption. In particular, F-PyraPVCNN ($0.25 \times C$) improves the detection rate of pedestrian by 4% compared with the F-PVCNN, and 7.5%

compared with the F-PointNet, validating the usefulness of PyraPVCNN in 3D detection tasks.

5. Conclusion

In this paper, we have presented a novel PyraPVConv block with the aim of efficient 3D point cloud perception. The key idea is to utilize voxel pyramid to make full use of voxel information and then adopt the attention mechanism for better point-voxel feature fusion. Extensive experiments validate the superiority of PyraPVConv. We hope that PyraPVConv can act as an important part of various deep networks for 3D point cloud perception. In the future, we plan to utilize the technique of neural architecture search for designing more efficient network architectures.

Data Availability

All datasets used in this paper are publicly available.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yuhong Chen and Weilong Peng contributed equally to this work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62102105, 62072126, and 61772164), Guangdong Basic and Applied Basic Research Foundation 2022A1515011501, 2022A1515010138 and 2020A1515110997 and Science and Technology Program of Guangzhou (202002030263 and 202102010419).

References

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] M. Frosi and M. Matteucci, "Art-slam: accurate real-time 6dof lidar slam," *IEEE Robotics and Automation Letters*, vol. 7, 2022.

[3] K. Xu, L. Zheng, Z. Yan et al., "Autonomous reconstruction of unknown indoor scenes guided by time-varying tensor fields," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–15, 2017.

[4] Y. Cui, R. Chen, W. Chu et al., "Deep learning for image and point cloud fusion in autonomous driving: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, 2021.

[5] Y. Zeng, Y. Hu, S. Liu et al., "Rt3d: real-time 3-d vehicle detection in lidar point cloud for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3434–3440, 2018.

[6] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: the principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[7] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3d local surface description and object recognition," *International Journal of Computer Vision*, vol. 105, no. 1, pp. 63–86, 2013.

[8] K. Tang, P. Song, and X. Chen, "Signature of geometric centroids for 3d local shape description and partial shape matching," in *Proceedings of the Asian Conference on Computer Vision*, pp. 311–326, Springer, Taipei, Taiwan, November 2016.

[9] K. Tang, P. Song, and X. Chen, "3d object recognition in cluttered scenes with robust shape description and correspondence selection," *IEEE Access*, vol. 5, no. 99, pp. 1833–1845, 2017.

[10] Y. Zhou and O. Tuzel, "Voxelnet: end-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, Salt Lake City, UT, USA, June 2018.

[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652–660, Honolulu, HI, USA, July 2017.

[12] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 4338–4364, 2020.

[13] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-yolo: an euler-region-proposal for real-time 3d object detection on point clouds," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 197–209, Munich, Germany, September 2018.

[14] B. Yang, W. Luo, and R. Urtasun, "Pixor: real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7652–7660, Salt Lake City, UT, USA, June 2018.

[15] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11873–11882, Seattle, WA, USA, June 2020.

[16] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, Long Beach, CA, USA, June 2019.

[17] M. Ye, S. Xu, and T. Cao, "Hvnet: hybrid voxel network for lidar based 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1631–1640, Seattle, WA, USA, June 2020.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 30, pp. 1–14, Long Beach, CA, USA, December 2017.

[19] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6411–6420, Seoul, Republic of Korea, November 2019.

[20] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 965–975, 2019.

[21] D. Maturana and S. Scherer, "Voxnet: a 3d convolutional neural network for real-time object recognition," in *Proceedings of the 2015 IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS), pp. 922–928, Hamburg, Germany, October 2015.

[22] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5648–5656, Las Vegas, NV, USA, June 2016.

[23] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: octree-based convolutional neural networks for 3D shape analysis," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–11, 2017.

[24] R. Klokov and V. Lempitsky, "Escape from cells: deep kd-networks for the recognition of 3d point cloud models," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 863–872, Venice, Italy, October 2017.

[25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[26] H. Tang, Z. Liu, S. Zhao et al., "Searching efficient 3d architectures with sparse point-voxel convolution," in *Proceedings of the European Conference on Computer Vision*, pp. 685–702, Springer, Glasgow, Scotland, August 2020.

[27] Q. Hu, B. Yang, L. Xie et al., "Randla-net: efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11108–11117, Seattle, WA, USA, June 2020.

[28] X. Zhu, H. Zhou, T. Wang et al., "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9939–9948, Montreal, Canada, October 2021.

[29] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918–927, Salt Lake City, UT, USA, June 2018.

[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, Lake Tahoe, Nevada, December 2012.

[32] J. Wang, S. Lan, M. Gao, and L. S. Davis, "Infofocus: 3d object detection for autonomous driving with dynamic information modeling," in *Proceedings of the European Conference on Computer Vision*, pp. 405–420, Springer, Glasgow, Scotland, August 2020.

[33] P. Hu, J. Ziglar, D. Held, and D. Ramanan, "What you see is what you get: exploiting visibility for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11001–11009, Seattle, WA, USA, June 2020.

[34] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 641–656, Munich, Germany, September 2018.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[36] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[37] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-fpn: multi-scale voxel feature aggregation for 3d object detection from lidar point clouds," *Sensors*, vol. 20, no. 3, p. 704, 2020.

[38] C. Bundesen, "A theory of visual attention," *Psychological Review*, vol. 97, no. 4, pp. 523–547, 1990.

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, Las Vegas, NV, USA, June 2016.

[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: visual explanations from deep networks via gradient-based localization," in *Proceedings of the International Journal of Computer Vision*, pp. 336–359, Venice, Italy, October 2017.

[41] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2019.

[42] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision*, pp. 3–19, Munich, Germany, September 2018.

[43] P. Anderson, X. He, C. Buehler et al., "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR*, pp. 6077–6086, Lake City, Utah, June 2018.

[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," pp. 1–15, 2014, https://arxiv.org/abs/1409.0473.

[45] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5998–6008, Long Beach California USA, December 2017.

[46] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: convolution on x-transformed points," in *Proceedings of the Advances in Neural Information Processing Systems*, vol. 31, pp. 820–830, Montréal, Canada, December 2018.

[47] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," pp. 1–9, 2017, https://arxiv.org/abs/1702.01105.

[48] I. Armeni, O. Sener, A. R. Zamir et al., "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1534–1543, Las Vegas, NV, USA, June 2016.

[49] A. X. Chang, T. Funkhouser, L. Guibas et al., "Shapenet: an information-rich 3d model repository," pp. 1–11, 2015, https://arxiv.org/abs/1512.03012.

[50] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*, pp. 424–432, Springer, Athens, Greece, October 2016.

[51] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 7839840, 21 pages
https://doi.org/10.1155/2022/7839840

Research Article

RLAS-BIABC: A Reinforcement Learning-Based Answer Selection Using the BERT Model Boosted by an Improved ABC Algorithm

Hamid Gharagozlou,[1] **Javad Mohammadzadeh**,[1] **Azam Bastanfard**,[1] **and Saeed Shiry Ghidary**[2]

[1]*Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran*
[2]*School of Digital, Technologies, and Arts, Staffordshire University, Stoke-on-Trent, UK*

Correspondence should be addressed to Javad Mohammadzadeh; j.mohammadzadeh@kiau.ac.ir

Received 23 January 2022; Revised 23 March 2022; Accepted 4 April 2022; Published 6 May 2022

Academic Editor: Ripon Chakrabortty

Answer selection (AS) is a critical subtask of the open-domain question answering (QA) problem. The present paper proposes a method called RLAS-BIABC for AS, which is established on attention mechanism-based long short-term memory (LSTM) and the bidirectional encoder representations from transformers (BERT) word embedding, enriched by an improved artificial bee colony (ABC) algorithm for pretraining and a reinforcement learning-based algorithm for training backpropagation (BP) algorithm. BERT can be comprised in downstream work and fine-tuned as a united task-specific architecture, and the pretrained BERT model can grab different linguistic effects. Existing algorithms typically train the AS model with positive-negative pairs for a two-class classifier. A positive pair contains a question and a genuine answer, while a negative one includes a question and a fake answer. The output should be one for positive and zero for negative pairs. Typically, negative pairs are more than positive, leading to an imbalanced classification that drastically reduces system performance. To deal with it, we define classification as a sequential decision-making process in which the agent takes a sample at each step and classifies it. For each classification operation, the agent receives a reward, in which the prize of the majority class is less than the reward of the minority class. Ultimately, the agent finds the optimal value for the policy weights. We initialize the policy weights with the improved ABC algorithm. The initial value technique can prevent problems such as getting stuck in the local optimum. Although ABC serves well in most tasks, there is still a weakness in the ABC algorithm that disregards the fitness of related pairs of individuals in discovering a neighboring food source position. Therefore, this paper also proposes a mutual learning technique that modifies the produced candidate food source with the higher fitness between two individuals selected by a mutual learning factor. We tested our model on three datasets, LegalQA, TrecQA, and WikiQA, and the results show that RLAS-BIABC can be recognized as a state-of-the-art method.

1. Introduction

Today, the questions charged in numerous domains in cyberspace, such as Stack Overflow and GitHub, are progressing quotidianly. QA is one of the vital branches of natural language processing (NLP) that can have the ability to answer questions automatically. QA can be made in two ways: Several methods focus on generating answers that usually employ developing networks like generative adversarial network (GAN) to create answers [1]. Nonetheless, they cannot guarantee accurate meaning and grammar. Another category of methods uses AS, one of the essential subtasks of QA which is also applied in other fields such as machine comprehension [2]. Over the last few years, the problem has been gaining an increasing amount of attention [3, 4]. A question q and a set of candidate answers $A = \{a_1, a_2, a_3, \ldots, a_N\}$ are given, and the goal is to attain $a_i \in A$ as the best answer to question q. Questions and answers can have various lengths, and multiple answers may be the true answer to a question.

From the literature, there are numerous methods for AS based on traditional and deep learning methods [5]. The

traditional approaches rely more on search engine [6], information retrieval [7, 8], handcrafted rules [9], or machine learning models [10, 11]. Information retrieval-based models work based on the keywords without using any semantic data, which makes it challenging to obtain the correct answers [12]. Handcrafted rule-based techniques cannot enfold all patterns, and their performance is delimited [13, 14]. In machine learning-based methods, features are manually made, so their quality laboriously depends on feature extraction [15, 16]. Some criteria and classifiers, including edit distance and support vector machine, consider the matching associations between AS pairs [11]. Typically, traditional methods suffer from two major weaknesses. First, they mostly do not use semantic information in keywords, features, or rules, causing them not to consider all-side relationships between QA pairs. Second, feature extraction and handmade rules are not flexible, leading to inferior generalization capability. After the appearance of deep learning, many problems in many domains [17–23], including AS, have been overshadowed by it. Deep learning-based methods for AS usually employ a convolutional neural network (CNN) [24] or LSTM to grab semantic features on various levels. The main task is to estimate the semantic similarity between a question-answer pair, which can be regarded as a text similarity calculation or classification work. A CNN is employed to model the hierarchical structures of sentences and evaluate their matching amount [25]. At the same time, an LSTM is considered to generate the embeddings of questions and answers while keeping sequential dependency information. Although deep models can only achieve limited improvement, they face some difficulties. They forge the embedding representation of the question-answer pair with one neural network design. This results in paying attention to one-side features and ignoring the other complex semantic features among question-answer couples. After that, models that try to comprehend languages were developed [26]. These models realize language syntactic and semantic rules in different methods, including next word and sentence prediction and masked word prediction [27]. They recognize a language and can make new texts with correct syntax and semantic rules. The BERT model [27] is one of the latest language models, being superior to all other developed language models. This model has grabbed advantage of the statement offered in transformers [28], which is currently widely employed in NLP tasks [29].

The success of deep models mainly relies on architecture, training algorithms, and selection of features employed in training. All these make the design of deep networks a complex optimization problem [30]. In many methods, the topology and transfer functions are set, and the space of possible networks is spanned by all potential values of the weights and biases [31]. In [32, 33] and [34], ant colony optimization [35], tabu search [36], simulated annealing [37], and genetic algorithm [38] were utilized for the training of neural networks with fixed topology. The neural network learning optimization process discovers the weight configuration associated with the lowest output error. Nevertheless, finding the optimal weight for deep models largely depends on weight initialization that has a more significant impact on neural network performance than network architecture and training examples [39]. AS methods, including in-depth ones, utilize gradient-based algorithms such as BP and Levenberg–Marquardt (LM) [40] for model weight optimization. While the BP algorithms converge in the first-order derivatives, the LM ones converge with second-order derivatives [41]. The main problem of BP and LM is the sensitivity to the initial weights, which leads to getting stuck in the local optimization [42]. To deal with this problem, global search approaches, having the power to evade local minima, are being employed to pretrain weights, such as population-based metaheuristic (PBMH) algorithms [43–45]. Among PBMH algorithms, ABC is one of the most powerful algorithms for optimization problems, which has two advantages over traditional algorithms: no need to calculate gradients and not getting caught up in local optimizations [46]. This algorithm is based on the intelligent behavior of bees, containing two general concepts: food sources and artificial bees. Artificial bees are looking for food sources with high nectar. The position of the food source shows a solution to the optimization problem, and the amount of nectar equals the quality of a solution. Although the food source position is a critical factor determining whether a bee selects a food source, some necessary information is still missing when bees produce a neighboring food source.

One of the other main problems in AS is imbalanced classes, since the member number of positive class, including the question and the corresponding answer, is much smaller than that of negative class, including the question and the non-corresponding answer, which reduces the performance of existing methods. Proposed methods with an imbalanced problem are generally divided into two groups: data-level methods and algorithmic-level methods. In data-level algorithms, training data is manipulated to balance class distribution by an oversampling minority class, undersampling majority class, or both. SMOTE [47] is an oversampling system that generates new examples by linear interpolation between adjacent minority samples. Near Miss [48] is an undersampling method that deals with an imbalanced problem by accidentally removing samples from a larger class. This algorithm eliminates the data of the larger class when viewing two data points belonging to two various classes that are close in terms of distribution. Oversampling algorithms can increase the possibility of overfitting, and undersampling algorithms lose valuable information in the majority class. In algorithmic-level methods, the importance of the minority class rises with techniques such as cost-sensitive learning, ensemble learning, and decision threshold adjustment. In the cost-sensitive learning methods, different costs are allocated to the wrong classification of each class in the loss function, which is more for the minority class. Ensemble learning-based solutions train multiple subclassifications and adopt voting to get better results. Threshold adjustment techniques train the classifier in the imbalanced dataset and change the decision threshold during the test. Deep learning-based methods have also been suggested to classify imbalanced data. The paper [49] introduced a loss

function for deep models that equally receives classification errors from the majority and minority classes. Another study in [50] learns the discriminative features of imbalanced data while maintaining intercluster and interclass margins. The authors in [51] presented a method based on the bootstrapping algorithm that balances training data of convolutional network per mini-batch. An algorithm is proposed by [52] for optimizing network weights and class-sensitive costs. In [53], the authors extracted complex samples in the minority class and improved their algorithm by batchwise optimization with Class Rectification Loss function [54].

In the last few years, deep reinforcement learning has been successfully used in computer games, robots' control, recommendation systems [55–57], etc. For classification problems, deep reinforcement learning has helped eradicate noisy data and learn better features, which significantly improved classification performance. Nonetheless, little research has been accomplished on applying deep reinforcement learning to imbalanced classification. Deep reinforcement learning is ideally appropriate for imbalanced classification as its learning mechanism, and specific reward function is comfortable paying more attention to minority class by giving higher rewards or penalties.

This paper presents an attention mechanism-based LSTM model for AS, called RLAS-BIABC, established on the BERT word embedding, reinforcement learning, and an improved ABC algorithm. The main body of the RLAS-BIABC model consists of two attention-mechanism-based bidirectional LSTM (BLSTM)networks and a feedforward network to calculate the similarity of the question-answer pair. The model aims to learn both positive and negative pairs. The positive pair is related to the question and real answer, while the negative one considers each question with the other answers. We use BERT as word embedding to learn the semantic similarity between sentences without pre-engineered features. What is more, we introduce an improved ABC algorithm for RLAS-BIABC, whose task is to find weight initialization in all LSTMs, the attention mechanism, and feedforward network to begin the BP algorithm. In this regard, we modify the ABC algorithm by applying mutual learning between two selected position parameters to choose the candidate food source with higher fitness. In addition, in the BP step, our proposed method employs reinforcement learning to handle imbalanced classification in the proposed method. In this respect, we define the AS problem as a guessing game divided into a sequential decision-making process. At each step, the agent takes an environmental state represented by a training instance and then executes a two-class classification operation under the guidance of a policy. If the classifier accomplishes the operation well, it will take a positive reward; otherwise, it will take a negative reward. The minority class is more rewarded than the majority one. The agent's goal is to get as many cumulative rewards as possible during the sequential decision-making process, that is, to classify the samples as accurately as possible. We assess the RLAS-BIABC model on three standard datasets, TrecQA, LegalQA, and WikiQA, and show RLAS-BIABC to be superior to other methods that use random weighting.

The main contributions of the article are as follows: (1) We consider the BERT word embedding, which is the last developed model for many languages. (2) Instead of using the random weight system for the model weights, we define an encoding strategy and compute an initial value using an improved ABC algorithm. (3) We consider the AS problem a sequential decision-making process and propose a deep reinforcement learning framework for imbalanced classification. (4) We study the performance of the proposed model through experiments and compare it with the other methods that use the random weight for initialization and are faced with the imbalanced classification problem.

The rest of this article is organized as follows: Section 2 presents a short review of related works. Section 3 introduces the ABC algorithm. Section 4 describes the framwork of the proposed model. Section 5 exhibits evaluation metrics, datasets, andresults. Section 6 provides a conclusion and future works.

2. Related Work

Until now, many approaches to the QA problem have been proposed. This section provides an overview of the methods based on machine learning and deep learning.

The first proposed approaches were based on feature engineering. In these methods, the relationship between question and answer is measured by repeating common words, where bag-of-words and bag-of-grams [58] are commonly applied for this purpose. These methods are not logical because they do not respect semantic and linguistic features in sentences. Subsequently, however, some studies have utilized language resources such as WordNet [59] to resolve the semantic problem but failed to remove linguistic limitations. Some researchers considered sentences' syntactic and semantic structure [60]. Some authors employed the dependency tree and the tree edit distance algorithm [15, 61]. The research [62] confirmed that tools such as WordNet and NER [63] could play an influential role in selecting semantic features. The article [64] provided an effective solution for automated feature selection. These methods were one of the first attempts to eliminate feature engineering.

Later, with the advent of deep learning, many methods used deep models as an automatic feature engineering tool. Recently, in-depth learning has covered a wide range of applications of NLP[18]. Moreover, recurrent neural network (RNN) and CNN are applied as two strong arms of deep learning in feature extraction [20, 21]. The behavior of deep learning methods with question-answer pairs is divided into two categories. In the first category, question and answer are two distinct elements, and deep networks reach their representation vectors separately. Typically, various criteria are adopted to measure the similarity between them. The authors in [65] offered a compare-aggregate system that applies many metrics for similarity measuring. The study [66] utilized the ELMo language model [26] to overcome question and answer work. The results reveal the superiority of language models. In the second category, question and answer are

assumed to be a single sentence. In [67], a CNN-based approach is presented to score question-answer pairs in a pointwise manner. Another technique in [68] applies the BLSTM network for question answering. Primarily, the embedding of question and answer words is learned and then entered into a BLSTM network, and later the embedding of each sentence is estimated based on the average of its words. Lastly, the answer-question connection is fed to a feedforward network. Siamese network [69] is an essential branch of in-depth learning that has been applied in all fields, especially QA. The network provides two separate representation vectors for question and answer. In [70], the first deep learning task is presented for the AS task. In this study, the most relevant answer to the question is extracted using a CNN and logistic regression. The research [71] implemented the idea presented in [70]. The authors tried to make different models using hidden layers, convolution operations, and activation functions to improve the results. Another work in [72] mixes various models to produce representation vectors for every sentence. In [73], the authors convert each point model into a pair model. Their idea was that pair models could further enhance model performance. The pair model was also applied to the model in [72]. The study [74] is a preprocessing operation. In this research, named entities are replaced with a unique token that facilitates selecting candidate answers. The impressive effectiveness of this technique was confirmed by applying it to the model presented in [73]. Meanwhile, the authors in [75] claimed that not all the named entities could be replaced with one token, so they considered a token for each named entity. It was later found that using the attention mechanism could produce more valuable models. Unlike the Siamese-based technique, the attention mechanism uses context-sensitive interactions [76] between question and answer. The attention mechanism was first proposed for machine translation but was later employed in other applications such as question answering [77, 78]. The approach in [79] considered the attention mechanism and RNNs to succeed in the answer-selection task. It was based on the attention mechanism proposed in [80]. In [81], the authors employed a method based on inter-weighted alignment networks to determine the similarity between a question-answer pair. The article [82] suggested a scheme based on a bidirectional alignment mechanism and stacked RNNs. In the first works, the attention mechanism was performed only on RNN, but later [83] pointed out that combining a CNN and attention mechanism could be more efficient.

3. Background

3.1. Long Short-Term Memory (LSTM). In a nutshell, RNNs [84] are designed to model sequential inputs. In these networks, a data sequence is mapped to a series of hidden states. The output is then generated using the following equations:

$$h_t = \theta\left(W_h h_{t-1} + U_h x_t + b_h\right). \tag{1}$$

$$y_t = \tau\left(W_y h_t + b_y\right), \tag{2}$$

where W_h and U_h are weight matrices and b means bias. θ and τ represent the activation functions such as ReLU and Tanh. $x_t \in \mathbb{R}^d$ is the input with dimension d, and $h_t \in \mathbb{R}^h$ equals the hidden layer with size h at time t.

RNNs have proven to be successful in many areas of NLP, such as text generation [85] and text summarization [86]. However, later, it became clear that as the length of the input of these networks increases, they suffer from problems such as gradient explosion and vanishing [87]. The LSTM network proposed by Hochreiter and Schmidhuber [88] can prevent the mentioned problems. This is because memory units can effectively handle long dependencies. In particular, LSTM consists of several control gates and one memory unit. Let x_t, h_t, and c_t represent input, hidden state, and memory cell at time t, respectively. Given a sequence of inputs $(x_1, x_2, \ldots, x_T)$, LSTM should calculate a sequence of hidden units $(h_1, h_2, \ldots, h_T)$ and memory cells $(c_1, c_2, \ldots, c_T)$ as output. In terms of formula, the specified process can be defined as follows [89]:

$$\begin{aligned}
i_t &= \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right),\\
f_t &= \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right),\\
c_t &= f_t c_{t-1} + i_t \tanh\left(W_j x_t + U_j h_{t-1} + b_j\right),\\
o_t &= \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right).\\
o_t &= \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right),\\
h_t &= o_t \tanh\left(c_t\right),
\end{aligned} \tag{3}$$

where W and b are network parameters. i, f, and o display input gate, forget gate, and output gate, respectively. σ stands for sigmoid function.

Although many problems can be solved under the umbrella of LSTM networks [18, 19, 90], experiments show that BLSTM can be more effective than LSTM. A BLSTM network [91] is an extended LSTM net that processes input from start to end and vice versa. This process generates two hidden vectors, $\overrightarrow{h}_t$ and h_t, for a specific input at the moment of t. Thus, the connected vectors, namely $[\overrightarrow{h}_t, h_t]$, form the final hidden vector.

The information extracted by the units in the LSTM network is equally important in making the final decision, which reduces system performance. To illustrate the point, consider the sentence "Despite being from Uttar Pradesh, as she was brought up in Bengal, she is convenient in Bengali." In this sentence, words like "Bengali" and "Bengal" should be given more attention, while this is not the case in an original LSTM network. To overcome this problem, the attention mechanism has been considered. In an attention mechanism system, the importance of each hidden layer with a coefficient in the interval [0, 1] is involved in the construction of the final vector. Formally, the hidden unit vector for a

particular input of length T is calculated by considering the coefficient α_t for each hidden vector h_t as follows:

$$h = \sum_{t=1}^{T} \alpha_t h_t. \tag{4}$$

3.2. Artificial Bee Colony (ABC) Algorithm. The ABC algorithm is a technique inspired by the intelligent behaviors of bees in nature. Two general concepts form the main body of the algorithm ABC: food sources and artificial bees. Artificial bees are looking for food sources with high nectar. The position of the food source indicates a solution to the optimization problem, and the amount of nectar corresponds to the quality of a solution. ABC involves three different groups of bees: employed, onlooker, and scout. Employed bees search for food sources with higher nectar in the vicinity of other food sources around them and share their information with onlooker bees in the dance area. The numbers of employed and onlooker bees are the same, and each is equal to half of the colony. Each employed bee exists in a hive, so the number of employed bees equals the total hives. Like employed bees, onlooker bees search for the best food sources in their neighborhood. Employed bees whose food resources do not improve after a few steps are converted to scouts, and a new search begins. The optimization process of ABC is summarized as follows:

Initialization Stage. Food sources as bee locations in the search space are initialized as follows:

$$x_i^j = x_{\min}^j + rand\,(0.1)\left(x_{\max}^j - x_{\min}^j\right). \tag{5}$$

where i refers to the i-th solution that takes the integer value in the interval $[1, BN]$, whereBN is the total number of solutions. Each solution consists of D elements, where D shows the number of weights to be optimized. $x_{\min}^j$ and $x_{\max}^j$ are the lowest and highest value in the solution i, respectively.

Employed Bee Stage. After initialization, the employed bees identify new sources in the neighborhood of existing food ones. Now they calculate the quality of the designated food sources. If their quality is better, they erase the information of previous sources from memory, replacing it with that of new sources. Otherwise, the data of earlier sources will remain unchanged. Formally, this step can be described by the following formula:

$$v_i^j = x_i^j + \varphi_i^j\left(x_i^j - x_k^j\right), \tag{6}$$

where k has an integer value in the interval $[1, BN]$, φ_i^j is a random decimal value in $[-1, 1]$, and v_i is a new food source derived from the change of an element x_i.

Onlooker Bee Stage. At this phase, the employed bees provide information to the onlooker bees. Onlooker bees calculate the value of the information and select the new solution based on the probability value. As in the previous step, if the new solution has more nectar, the previous position information will be replaced with the new solution. The possibility of choosing a new solution can be formulated as follows:

$$p_i = \frac{\mathrm{fit}\,(x_i)}{\sum_{n=1}^{BN} \mathrm{fit}\,(x_n)}, \tag{7}$$

where $\mathrm{fit}\,(xi)$ is the fitness value for the i-th solution. According to (7), the higher the $\mathrm{fit}\,(xi)$ is, the more likely the observer bee will accept this solution. The onlooker bee goes to it if the selection is performed, and a new solution is generated according to (6).

Scout Bee Stage. In the last step, scout bees are employed to escape the local optimum. More specifically, any solution that fails to improve the process after some cycles becomes a scout bee, and the food source is dropped. Therefore, a new food source replaces the old one according to (6).

The four steps mentioned above are performed up to several times to meet the termination criteria. The complete ABC algorithm is given in Algorithm 1.

4. The Framework of RLAS-BIABC

The proposed algorithm considers two critical options for classification. In the first step, we formulate a vector that includes all the learnable weights in our model, and we optimize it utilizing the ABC algorithm. Then, we apply the BP algorithm in the rest of the path. Besides, another problem that most classifiers suffer from, including ours, is imbalanced data. To take this aspect into account, we employ the opinions of reinforcement learning. We present these two ideas in two separate sections.

The general architecture of the proposed model is shown in Figure 1. Consider a question Q containing a sequence of n words, where $Q = (q_1, q_2, \ldots, q_n)$, with the answer A, where $A = (a_1, a_2, \ldots, a_m)$ including m words. Let $a_i, q_j \in \mathbb{R}^D$ show the D-dimensional visual presentations of a word. Two LSTMs are provided for each question and answer. Two pairs of positive and negative data are used to learn the model. In the positive pair (Q, A), A is the correct answer to question Q; the output of the model should go to one. Meanwhile, in the negative pair (Q, A'), where A' is the fake answer to the question, the network should move to zero for this pair. The embedding calculated by LSTMs for question and answer is expressed as follows:

$$\begin{aligned} q &= \sum_{i=1}^{n} \alpha_i h_{q_i}, \\ a &= \sum_{i=1}^{m} \beta_i h_{a_i}, \end{aligned} \tag{8}$$

where $h_{q_i} = [\overleftarrow{x}_i, \overrightarrow{x}_i]$ and $h_{a_i} = [\overleftarrow{y}_i, \overrightarrow{y}_i]$ are the output of i-th BLSTM related to the question and answer, respectively. α_i and β_i are the attention weights of each unit that are computed as follows:

Input: D: dimensions of the solution, BN: population size, limit: number of cycles, MaxItr: maximum number of iterations
(1) Initialize the population of solutions $X = [x_1, x_2, \ldots, x_{BN}]$ using (5)
(2) Itr = 1.
(3) **while** ≤ MaxItr**do**
(4) //Employed Bee Phase
(5) **for**i = 1 to BN**do**
(6) Produce new solution x_{new} using (6)
(7) Calculate the fitness f_{new} for x_{new}
(8) Replace x_{new} with x_i if better
(9) **end for**
(10) Calculate the probability p for every solution in X using (7)
(11) //Onlooker Bee Phase
(12) **for**i = 1 to BN**do**
(13) **if** rand $(0, 1) < p_i$**then**
(14) Produce new solution x_{new} using (6)
(15) Calculate the fitness f_{new} for x_{new}
(16) Replace x_{new} with x_i if better
(17) **end if**
(18) **end for**
(19) //Scout Bee Phase
(20) If an abandoned solution is found, replace it with the solution produced by (6)
(21) Put the best solution ever in x_{best}
(22) Itr = Itr + 1.
(23) **end while**
(24) **return**x_{best}

Algorithm 1: Pseudocode of the ABC algorithm.

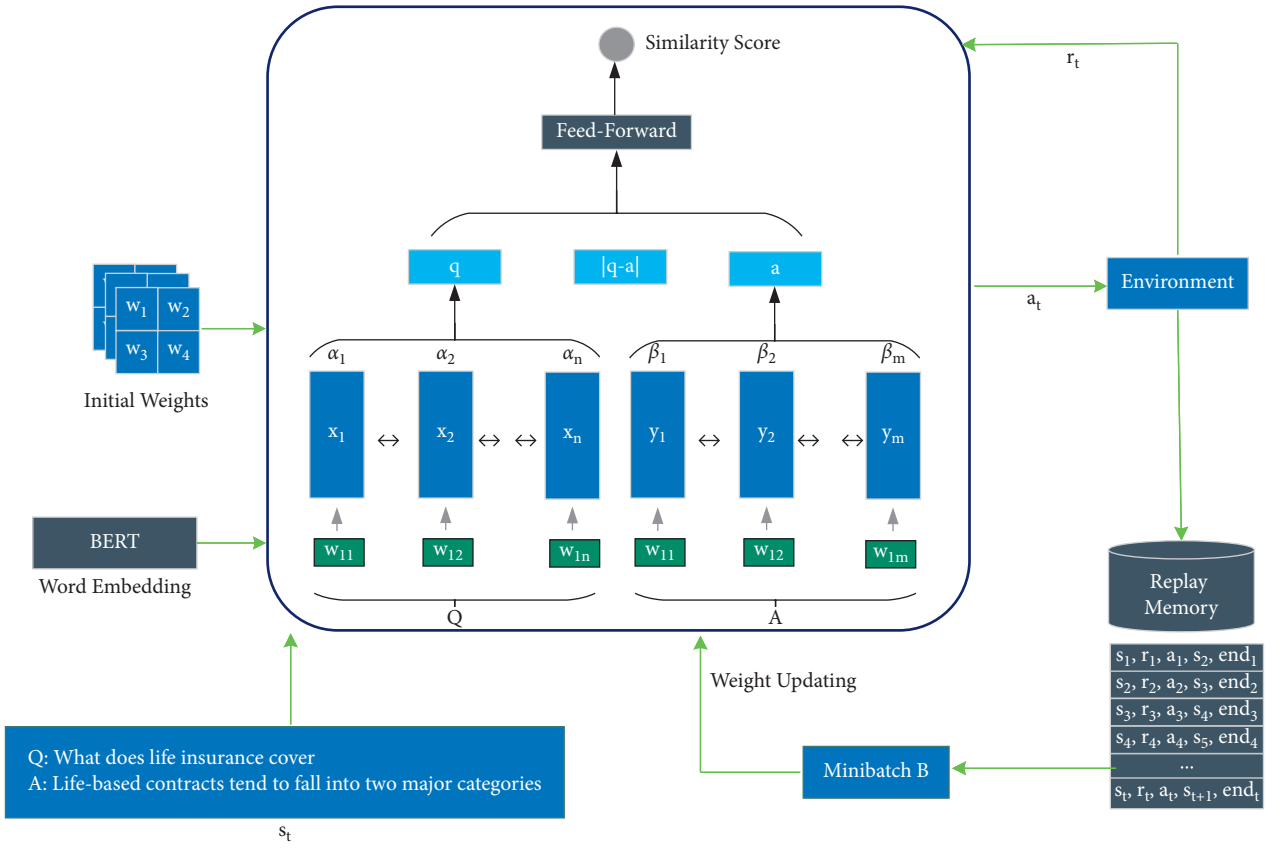


Figure 1: The proposed LSTM-similarity model.

$$\alpha_i = \frac{e^{u_i}}{\sum_{i=1}^{n} e^{u_i}},$$
$$\beta_i = \frac{e^{v_i}}{\sum_{i=1}^{m} e^{v_i}},$$
$$u_i = \tanh\left(W_u\left[\overleftarrow{x}_i, \overrightarrow{x}_i\right] + b_u\right), \quad (9)$$
$$v_i = \tanh\left(W_v\left[\overleftarrow{y}_i, \overrightarrow{y}_i\right] + b_v\right),$$

where W_u, W_v, b_u, b_v represent the parameters of the attention mechanism. After determining the efficient representation of question and answer by the attention mechanism, we form a vector consisting of the connected q, a, and $|q - a|$ according to Figure 1 and enter it into a feedforward network. It has been experimentally confirmed that the difference between two representation vectors can act in a successful decision [92].

4.1. BERT-Based Word Embedding. Word embedding serves as a function of mapping words to semantic vectors for use in deep learning algorithms. Word embedding is a reliable way to extract significant representations of words established in their context. Much research has been conducted to find the best meaningful word representations on neural network models such as Skip-gram [93], GloVe [94], and FastText [95]. Over the last few years, the pretrained language model (PLM), which is a black box with prior knowledge of the natural language and is fine-tuned in NLP works, has been much applied.

PLM models generally use unlabeled data to learn model parameters [96]. The paper considers the BERT model [27], one of the latest techniques in the PLM trends. BERT is a bidirectional language model trained on big datasets such as Wikipedia to generate contextual representations. In addition, it is commonly fine-tuned from a neural network dense layer for different classification duties. The fine-tuning functionality includes the contextual or the problem-specific meaning with the pretrained generic meaning and trains it for a classification task.

Figure 2 indicates the architecture of a BERT model. BERT uses a bidirectional transformer, in which its representations are jointly conditioned on both the left and right context in layers [97], which differentiates it from the other models, including Word2Vec and GloVe, that build an embedding in one direction to dismiss its contextual differences.

4.2. Pretraining Stage. Weight initialization is an essential point in designing a neural network, the nonobservance of which leads to misleading the model. The proposed structure has two LSTM networks, two attention mechanisms, and one feedforward neural network, each of which has its weights that must be trained. The paper uses an improved ABC algorithm for pretraining weights.

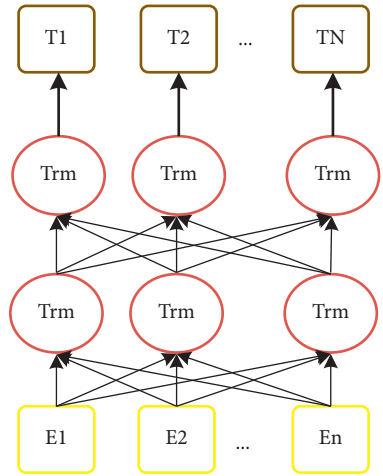


Figure 2: Architecture of the BERT model.

4.2.1. Mutual Learning-Based ABC. In the standard ABC algorithm, artificial bees randomly select a food source position and change it to create a new position. If the fitness value of the new solution is better, it will replace the current solution. Otherwise, no change will be applied. In other words, in a D-dimensional optimization problem, one dimension is randomly selected, its value is changed, and the better outcome is selected in each iteration. Based on (6), the newly generated solution v_i^j depends on only two parameters, x_i^j and x_k^j, making the food source v_i^j uncontrollable, sometimes larger and sometimes smaller than the current food source. In the ABC algorithm, a food source with a higher fitness value is required. To always produce a food source a higher value, we consider the fitness information acquired by mutual learning between current and neighboring food sources.

$$v_i^j = \begin{cases} x_i^j + \varphi_i^j\left(x_k^j - x_i^j\right), & \text{Fit}_i < \text{Fit}_k \\ x_k^j + \varphi_i^j\left(x_i^j - x_k^j\right), & \text{Fit}_i \geq \text{Fit}_k \end{cases}, \quad (10)$$

where Fit_i and Fit_k indicate the fitness value of the current food source and the neighboring food source, respectively. φ_i^j shows a uniform random number in the interval $[0, F]$, in which F is a nonnegative constant named the mutual learning factor. As we can see, the value v_i^j depends on their position and their value of fitness. By comparing the current and neighboring food sources, the fitness values of new solutions move to better sources. That is, if the current food source has higher suitability, the candidate solution will move toward a better solution; otherwise, it will tend to move toward the neighboring source. This learning strategy allows making a better candidate solution. The parameter F plays an essential role in balancing the perturbation between related food positions. In addition, F must be a nonnegative positive number to ensure it goes to a better solution. As F increases from zero to a particular value, the perturbation on the corresponding position decreases, meaning that the fitness value of the new food source is close to the higher fitness. A large value of F weakens the power of exploitation and exploration.

4.2.2. Encoding Strategy. Encoding means the weights are arranged in a vector, which is considered the bees' position in ABC. Choosing the right layout is a challenging task; however, we tried to design the best encoding strategy possible after several experiments. Figure 3 denotes an example of the encoding for two LSTMs, two attention mechanisms, and a two-layer feedforward network. Note that all weight matrices are stored in rows.

4.2.3. Fitness Function. The purpose of the fitness function is to measure the efficiency of a solution. The paper employs the following function as a competency function:

$$\text{fitness} = \frac{1}{1 + \sum_{i=0}^{T} \left(y_i - \widetilde{y}_i\right)^2}, \tag{11}$$

where T is the total number of training samples and y_i and $\widetilde{y}_i$ are the target and predicted labels for the i-th data, respectively.

4.3. Classification. Reinforcement learning (RL) [98] is a subfield of machine learning that solves a problem by making successive decisions [99, 100]. Recently, reinforcement learning has achieved excellent results in classification because it can learn valuable features or select high-level samples from noise data. In [101], the classification problem was defined as a sequential decision-making process that used several factors to learn the optimal policy. However, complex simulations between agents and environments have somewhat increased the time complexity. Another work in [102] submitted a solution for learning a relationship in text noise data. For this purpose, the proposed model is divided into two parts: instance selector and relational classifier. The instance selector is designed to extract quality sentences from noise data with the agent help. At the same time, the relational classifier learns better performance from selected clean data and gives delayed reward feedback to the instance selector. Finally, the model results in a better classification and quality dataset. The authors in [103–106] considered deep reinforcement learning to learn the helpful training data features. Generally, they improved the valuable features of the classifier. The work in [107] used reinforcement learning to classify time series data in which the reward function and the Markov model are designed. So far, little research has been done on the classification of unbalanced data, especially the processing of natural languages using reinforcement learning. In [108], an ensemble pruning method that picks the best sub-classifiers under the reinforcing learning umbrella was developed. This method was effective for small data because it was practically impossible to choose classifiers with many subcategories.

This section describes how to apply reinforcement learning to prevent imbalanced classification. Overall, the agent receives a sample at each step and classifies it. After that, the environment gives immediate and next rewards to the agent. A positive reward is assigned to the agent by the environment when it categorizes the sample correctly. Otherwise, it receives a negative reward. Finally, the agent learns the optimal behavior by maximizing the aggregate rewards and then can classify the samples as accurately as possible.

Let $D = \{(x_1, l_1), (x_2, l_2), \ldots, (x_T, l_T)\}$ be training data, where $x_i = (q_i, a_i)$ is the$i$-th sample so that q_i and a_i are the i-th question and answer that enter the model, respectively. $l_i \in \{0, 1\}$ shows the target of the i-th example. We consider the following conditions for an agent.

4.3.1. Policy π_θ. The policy π_θ is a mapping function π: $S \longrightarrow A$ where $\pi_\theta(s_t)$ denotes the action a_t performed by an agent in state s_t. In our work, the proposed classification with the set weight θ is recognized as policy π_θ.

4.3.2. State s_t. Each example of the training dataset is described as a state. The agent takes the first data x_1 as the initial state s_1 at the start of the training. State s_t at each time step t corresponds to x_t in the training dataset. The order of the samples in each iteration is different for the agent.

4.3.3. Action a_t. The action performed by the agent is to predict the category label. Hence, the agent's performance is related to the training dataset label. The recommended model is a binary classifier, $a_t \in \{0, 1\}$, where zero and one show the minority and majority classes, respectively. In this context, the relevant question and answer are one, and the irrelevant question and answer are zero.

4.3.4. Reward r_t. The agent receives a positive score if the sample is classified correctly and a negative score otherwise. Since minority class instances are more critical because of their small number, the algorithm should consider the size of the score for the minority class more. The reward function is described as follows:

$$r\left(s_t, a_t, l_t\right) = \begin{cases} +1, a_t = l_t \text{ and } s_t \in D_P \\ -1, a_t \neq l_t \text{ and } s_t \in D_P \\ \lambda, a_t = l_t \text{ and } s_t \in D_N \\ -\lambda, a_t \neq l_t \text{ and } s_t \in D_N \end{cases}, \tag{12}$$

where $\lambda \in [0, 1]$, and D_P and D_N are related to the minority and majority classes, respectively. l_t is the label of the sample x_t. The bonus amount is considered the cost of predicting the label. According to this relation, when $\lambda < 1$, the amount of the cost of the minority class is more. If the distribution of all classes is balanced, $\lambda = 1$, then the prediction cost of all classes is the same. We will examine the different values of λ in our experiments.

4.3.5. Terminal E. The episode is a transition trajectory from the initial state to the terminal state $\{(s_1, a_1, l_1), (s_2, a_2, l_2), \ldots, (s_t, a_t, l_t)\}$. An episode finishes when all instances in the training data are classified or when the agent misclassifies the instance from the minority class.

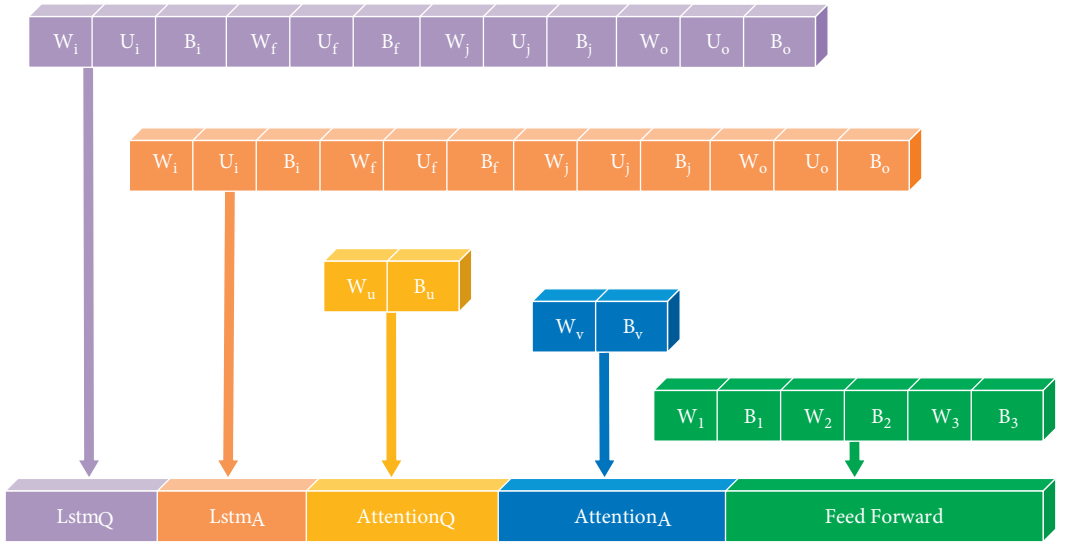


Figure 3: Placement of weights in a vector.

4.3.6. Transition Probability P. The model transition probability, i.e., $p(s_{t+1}|s_t, a_t)$, is deterministic. The agent transfers from state s_t to state s_{t+1} according to the order of instances in the dataset.

In the proposed model, the π policy takes the input data and calculates its label probability:

$$\pi(a \mid s) = P \cdot (a_t = a \mid s_t = s). \tag{13}$$

The agent aims to identify the data input sample as accurately as possible. The best performance is attributed to the agent when it can maximize its cumulative rewards as follows:

$$g_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}. \tag{14}$$

Equation (14) is called the return function, the total accumulated return from time t with the discount factor $\gamma \in (0, 1]$ until the time when the agent moves in the search space. The action value Q in RL expresses the expected return for action a in state s, which can be defined as follows:

$$Q^{\pi}(s, a) = E_{\pi}[g_t \mid s_t = s, a_t = a]. \tag{15}$$

Equation (15) can be extended according to the Bellman Equation [109]:

$$Q^{\pi}(s, a) = E_{\pi}[r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) \mid s_t = a, a_t = a]. \tag{16}$$

By maximizing function Q under policy π, we can maximize cumulative rewards, namely Q^*. The optimal policy π^* obtained under function Q^*, which is a policy that performs best for our model, is as follows:

$$\pi^*(a \mid s) = \begin{cases} 1, & a = \operatorname{argmax}_a Q^*(s, a) \\ 0, & \text{else} \end{cases}. \tag{17}$$

By combining (16) and (17), function Q^* is computed as follows:

$$Q^*(s, a) = E_{\pi}[r_t + \gamma \max_a Q^*(s_{t+1}, a_{t+1}) | s_t = a, a_t = a]. \tag{18}$$

For low dimensions, the values of the function Q are collected in a table to obtain the optimal value according to the recorded values. However, the function Q can no longer be solved when the dimensions of the problem are continuous. To solve this problem, a deep Q-learning algorithm was adopted to model the function Q with a deep neural network. To that end, the tuple (s, a, r, s') obtained from (18) is stored in replay memory M. The agent selects a mini-batch B of transitions from M randomly and executes the dissent gradient algorithm on the deep Q network according to the following loss function:

$$L(\theta_k) = \sum_{(s,a,r,s') \in B} (y - Q(s, a; \theta_k))^2, \tag{19}$$

where y is the prediction of the function Q, which is formulated as follows:

$$y = \begin{cases} r, & \text{end} = \text{True} \\ r + \gamma \max_{a'} Q(s', a'; \theta_{k-1}), & \text{end} = \text{True} \end{cases}, \tag{20}$$

where s' indicates the next state s, and a' is the action executed in state s'.

4.4. Overall Algorithm. We design the simulation environment according to the contents defined above. The network architecture of the policy largely depends on the complexity and number of training examples. In this context, the input of the network depends on the structure of the training samples, and the output is equal to the number of classes of instance data. The general training algorithm of the model presented in Algorithm 2 is shown. First, the initial weights of the policy π are initialized using the ABC algorithm, and then the agent continues the training process until the optimal policy is reached. The choice of action is made based

on the greedy policy, and the selected action is evaluated by Algorithm 3. The algorithm is repeated E times, where E in this paper is considered 15,000. At each step, the policy network weights are stored.

5. Results

5.1. Datasets. A dataset with many negative pairs can be one of the best options to evaluate the performance of the proposed system. We run our experiments on three datasets, LegalQA, TrecQA, and WikiQA, which are widely considered by many researchers. All three datasets have more negative than positive pairs. The statistical information of all datasets is shown in Table 1:

(i) TrecQA [110] is derived from TREC track data. Yao et al. [10] made a complete version of the positive and negative pair set. Two training datasets, TRAIN and TRAIN-ALL, are available in this database. The correctness of the answers in TRAIN-ALL is checked automatically by matching pairs with regular expressions. All answers in the TRAIN, DEV, and TEST data were judged manually. We employ the TRAIN-ALL data to train our model.

(ii) LegalQA [111] is a Chinese dataset of legal consultative questions collected from a Chinese association. Users' online questions have been answered by licensed lawyers. LegalQA includes four fields: question subject, question body, answer, and label. The positive pair is provided as ground truth directly online.

(iii) WikiQA [112] is an open-domain QA dataset in which each question is linked to a Wikipedia page that is assumed to be the topic of the year. To eliminate answer sentence prejudice, all answers in the summary section of the page are considered candidate answers.

5.2. Evaluation Metrics. According to previous research, MAP and MRR are the most common criteria for evaluating answer-selection tasks [77]. MAP measures the ability to rank answers to return the corresponding answer. However, MRR is repeated if a high-scoring match is found:

(i) MAP (mean average precision) calculates the mean average precision on the ranking results as follows:

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n_i}\sum_{j=1}^{n_i}\mathrm{precision}\left(R_{ij}\right), \tag{21}$$

where Q denotes the set of questions, n_i is the number of answers to the i-th question, and R_{ij} means the set of ranked results to question j from the best result to the j-th answer.

(ii) MRR (mean reciprocal rank) evaluates the model suitability according to the position of the first correct answer, computed as follows:

$$\mathrm{MRR}(Q) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{r_i}, \tag{22}$$

where r_i indicates the position of the first matching answer for the i-th question.

5.3. Baseline Methods. We evaluate our RLAS-BIABC model with several state-of-the-art methods for answer selection. The following are the details of these methods:

KABLSTM [113] is a knowledge-aware method based on attentive BLSTM networks. This method uses knowledge graphs (KG) to learn the representation of questions and answers.

EATS [75] adopted an RNN network to measure the similarity between the QA pair. First, it replaces each named entity with a specific word. This system calculates sentence representation vectors by the attention mechanism. Finally, these vectors are entered into the feedforward network, and the similarity is calculated by the sigmoid function in the last layer.

AM-BLSTM [114] considered two LSTM networks for a question and answer separately. The resulting embeddings were combined and entered into an multilayer perceptron (MLP) network for classification. Moreover, traditional techniques, such as penalties for each class, have been employed to prevent imbalanced classification.

BERT-Base [115] introduced a search engine and transformer model method for selecting answers. This article adopts simple models such as Jaccard similarity and compare-aggregate to rank the answers to a question.

DRCN [116] offered an architecture based on a densely connected recurrent and co-attentive network in which hidden features are maintained at the top layer. Connection operations in this paper are performed using the attention mechanism to preserve information better. In addition, an autoencoder has been adopted to reduce the volume of information.

P-CNN [117] introduced a new approach using a positional CNN for text matching that considers positional information at the word, phrase, and sentence levels.

DARCNN [118] combined BLSTM, self-attention, cross-attention, and CNN to find the global and local features of the question and candidate answer, leading to better semantic modeling. Finally, it utilizes an MLP network to assign a score to a question-answer pair.

DASL [119] submitted a model with a Bayesian neural network (BNN) to effectively optimize the loss in the ranking learning process. Another study of this article is how to combine active learning and self-paced learning for model training.

KAAS [120] applied an interactive knowledge-enhanced attention network for AS that extracts rich features of question and answer knowledge at several levels. Additionally, an attention and self-attention network is considered to learn the semantic features of sentences.

Input: $D = \{(x_1, l_1), (x_2, l_2), \ldots, (x_T, l_T)\}$: a training dataset of size T
Initialize the weights of policy π using Algorithm 1
Initialize environment ε
Initialize replay memory M
for episode $e = 1$ to E**do**
  Shuffle the dataset D
  $s_1 = x_1$
  for$t = 1$ to T**do**
    $a_t = \pi_0(s_t)$ //select an action based on ε-greedy
    $[r_t, \text{end}_t] = \text{Reward}(x_t, a_t, l_t)$
    $s_{t+1} = x_{t+1}$
    Save $(s_t, a_t, r_1, s_{t+1}, \text{end}_t)$ to M
    Sample randomly a mini-batch of transitions (s_k, a_k, r_k, s_{k+1}) from M
    $y_k = \begin{cases} r_k, & \text{end}_k = \text{True} \\ r_k + \gamma \max_{a'} Q(s_{k+1}, a'; \theta), & \text{end}_k = \text{False} \end{cases}$
    Accumulate gradients w.r.t θ: $d\theta = d\theta + \partial(y_k - Q(s, a; \theta))^2/\partial\theta$
    ifend_t = True**then**
      break
    end if
  end for
end for

Algorithm 2: Pseudocode for training RIAS-BIABC.

Function Reward (x_t, a_t, l_t):
  end_t = False
  if$x_t \in D_p$**then**
    if$a_t == l_t$**then**
      $r_t = 1$
    else
      $r_t = -1$
      end_t = True
    end if
  else:
    if$a_t == l_t$**then**
      $r_t = \lambda$
    else:
      $r_t = -\lambda$
    end if
  end if
  return$[r_t, \text{end}_t]$

Algorithm 3: Pseudocode of reward function.

Table 1: Statistical information of LegalQA, TrecQA, and WikiQA datasets.

Dataset (TRAIN/DEV/TEST)	# questions	# QA pairs	% correct
LegalQA	10,526/1,593/3,035	100,590/11,965/26,913	21.8/24.4/22.9
TrecQA	1,229/82/100	53,417/1,148/1,517	12.0/19.3/18.7
WikiQA	873/126/243	20,360/1,130/2,352	12.0/12.4/12.5

"% correct" means the proportion of matched QA pairs.

5.4. Details of Implementation. In this work, Python and PyTorch have been utilized for the implementation. Jupyter has been used to implement project codes. Another library used in this study is NLTK. This library provides classes and methods for processing natural languages in Python. This library can perform a wide range of NLP operations. We use a two-layer BLSTM. Moreover, due to the connection of vectors in the two networks, we employ batch normalization before the data enters the feedforward neural network. Table 2 indicates the values of the other parameters.

Our project uses a 64-bit Windows operating system with 64 GB of RAM and GPU. The best model was obtained for the LegalQA, TrecQA, and WikiQA after 50, 60, and 100 epochs, respectively. The whole process of our training took 5, 20, and 60 hours for the three datasets.

5.5. Experimental Results. Due to heuristic algorithms working randomly, we repeated all the experiments 10 times. Quantitative results of the three datasets are given in Table 3. In addition to comparing the proposed method with the state-of-the-art algorithms, to evaluate the effectiveness of ABC and RL components on the model, we employ three techniques: AS + random weight, AS-BIABC, and RLAS. AS + random weight is a system applying only random weights for initial weighting. Models AS-BIABC and RLAS accept only ABC and RL, respectively. For the LegalQA dataset, the RLAS-BIABC model has beaten other models, including IKAAS, in the MAP and MRR criteria, so that our model has reduced the error by more than 40% and 24% in these two criteria. By comparing RLAS-BIABC with AS-BIABC and RLAS, we can see that it decreases the error rate by about 51%, indicating the importance of the initialization and RL approaches. For the TrecQA dataset, our algorithm obtained the highest MAP and MRR, followed by EAT algorithm. The error improving rate in this database is approximately 30.13% and 21.00% for MAP and MRR criteria, respectively. In the WikiQA dataset, RLAS-BIABC decreases the classification error by more than 32% and 42% compared to IKAAS and DRCN, respectively.

Next, we prove that the improved ABC is more powerful than others. To do this, we fix all pieces of our algorithm for a fair comparison, including the LSTM networks, the attention mechanisms, and the reinforcement learning, and only change the trainer. To reach this goal, we compare our offered trainer with six conventional algorithms, including GDM [121], GDA [122], GDMA [123], OSS [124], and BR [125], and eight metaheuristic algorithms, including GWO [126], BAT [127], DA [128], SSA [129], COA [130], HMS [131], WOA [132], and ABC [133]. In all metaheuristic methods, population size and function evaluations are 100 and 3,000, respectively. The rest of the parameters of the algorithms are shown in Table 4. The results of metaheuristic and conventional algorithms are collected in Table 5. RLAS-AM-BR and RLAS-BABC performed best for all datasets for conventional and metaheuristic algorithms. As we expected, the metaheuristic algorithms perform better than the conventional ones. Without exaggeration, the improved ABC has a more acceptable performance than all of them, so that compared to the best algorithm, i.e., the main version of ABC, it can diminish the error by approximately 16%.

5.5.1. The Effect of the Reward Value of Majority Class. The environment helps the agent achieve the goal by considering the reward function. This article considers two different rewards for the minority and majority classes. Minority class reward was set to $+1/-1$ while the majority class was set to $+\lambda/-\lambda$. To investigate the effect of the value of λ on the proposed model, we test it with the values in the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. The results of this experiment for the three datasets are indicated in Figure 4. As we see, for the LegalQA dataset, when λ has a value in the range $[0, 0.4]$, we have an uptrend, while we have a downtrend for the values $(0.4, 1]$. Hence, we fixed the value of λ for this dataset to 0.5. The best value of λ for both TrecQA and WikiQA datasets is 0.5. Generally, as the dataset size increases, the number of negative pairs increases, so λ tends to decrease. For $\lambda = 0$, the importance of the majority class is overlooked, and for most $\lambda = 1$, the importance of both classes is equal.

Table 2: The parameters of the model.

Parameter	Value
Batch size	128
Embedding dim	60
Max sentence length	80
Activation function (LSTM and dense)	ReLU
Dense hidden layer	8

5.5.2. Exploration on Loss Function. Traditional techniques, including manipulating the loss function and data augmentation, can also deal with data imbalances. However, they largely depend on the issue at hand. In the meantime, the loss function has a more colorful role because it can make the minority class more prominent. To check the inefficiency of the loss functions on our model, we selected the five functions Weighted Cross-Entropy (WCE) [134], Balanced Cross-Entropy (BCE) [135], Focal Loss (FL) [136], Dice Loss (DL) [137], and Tversky Loss (TL) [138]. The WCE and BCE loss functions give weight to the positive and negative samples. The FL function is suitable for applications with imbalanced data. It downweights the contribution of uncomplicated examples and allows the model to focus more on learning complex samples [139]. The evaluation results of these loss functions for the three datasets are shown in Table 6. The results show that all the functions have about the same MRR and MAP in the three datasets. As expected, the FL function performs better than the others, so it is about 51.16% better than the algorithm with the usual loss function, i.e., the RLAS-BABC model.

5.5.3. Case Study. In this section, we intend to qualitatively evaluate the effectiveness of reinforcement learning in our model. For this purpose, we randomly select a sample from the TrecQA dataset. Given the question, "When were the Nobel Prize awards first given?" top answers are given in Table 7. The left column presents the model results without using reinforcement learning, and the right column shows the model results with reinforcement learning. Our results say that the model without reinforcement learning is more inclined to assign a higher score to negative responses. However, the model with reinforcement learning has assigned as many scores as possible to the answers to the question.

Table 3: The evaluation results of the proposed model and other models.

Method	Dataset					
	LegalQA		TrecQA		WikiQA	
	MAP	MRR	MAP	MRR	MAP	MRR
KABLSTM [113]	0.751	0.790	0.792†	0.844†	0.732†	0.749†
EATS [75]	0.778	0.810	0.854†	0.881†	0.700†	0.715†
AM-BLSTM [114]	0.786	0.836	0.818	0.827	0.780	0.788
BERT-Base [115]	0.838	0.850	0.823	0.812	0.813†	0.828†
DRCN [116]	0.828	0.859	0.802	0.832	0.804†	0.862†
P-CNN [117]	0.715	0.729	0.680	0.698	0.734†	0.737†
DARCNN [118]	0.700	0.745	0.743	0.725	0.734†	0.750†
DASL [119]	0.804	0.816	0.824	0.831	0.768	0.795
IKAAS [120]	0.825†	0.883†	0.823†	0.868†	0.835	0.849
AS + random weight	0.758 ± 0.000	0.801 ± 0.001	0.796 ± 0.000	0.806 ± 0.002	0.771 ± 0.002	0.792 ± 0.009
AS-BIABC	0.788 ± 0.012	0.815 ± 0.008	0.802 ± 0.005	0.826 ± 0.002	0.803 ± 0.000	0.845 ± 0.025
RLAS	0.855 ± 0.102	0.872 ± 0.018	0.862 ± 0.014	0.883 ± 0.150	0.852 ± 0.025	0.876 ± 0.026
RLAS-BIABC	0.895 ± 0.020	0.912 ± 0.001	0.898 ± 0.015	0.906 ± 0.092	0.888 ± 0.036	0.891 ± 0.017

† indicates that the results are taken from the articles.

Table 4: Parameter setting of experiments.

Algorithm	Parameter	Value
ABC	Limit	$n_e \times$ dimensionality of problem
	n_o	50% of the colony
	n_e	50% of the colony
	n_s	1
GWO	No parameters	
BAT	Constant for loudness update	0.4
	Constant for an emission rate update	0.6
	Initial pulse emission rate	0.002
DA	Scaling factor	0.3
	Crossover probability	0.7
SSA	No parameters	
COA	Discovery rate of alien solutions	
HMS	Number of clusters	5
	Minimum mental processes	2
	Maximum mental processes	5
WOA	C	1
	B	1

Table 5: The performance of other methods for initialization.

Method	Dataset					
	LegalQA		TrecQA		WikiQA	
	MAP	MRR	MAP	MRR	MAP	MRR
RLAS-BGDM	0.796 ± 0.002	0.819 ± 0.026	0.824 ± 0.093	0.836 ± 0.026	0.810 ± 0.056	0.825 ± 0.136
RLAS-BGDA	0.783 ± 0.125	0.776 ± 0.095	0.769 ± 0.025	0.786 ± 0.269	0.745 ± 0.136	0.761 ± 0.002
RLAS-BGDMA	0.791 ± 0.005	0.772 ± 0.103	0.796 ± 0.126	0.812 ± 0.236	0.793 ± 0.026	0.793 ± 0.005
RLAS-BOSS	0.810 ± 0.136	0.814 ± 0.004	0.853 ± 0.023	0.863 ± 0.026	0.840 ± 0.027	0.855 ± 0.127
RLAS-BBR	0.842 ± 0.009	0.853 ± 0.000	0.860 ± 0.036	0.878 ± 0.120	0.852 ± 0.103	0.870 ± 0.035
RLAS-BGWO	0.771 ± 0.205	0.783 ± 0.018	0.755 ± 0.072	0.781 ± 0.126	0.755 ± 0.025	0.773 ± 0.026
RLAS-BBAT	0.862 ± 0.003	0.818 ± 0.019	0.876 ± 0.093	0.880 ± 0.239	0.852 ± 0.061	0.873 ± 0.082
RLAS-BDA	0.816 ± 0.072	0.829 ± 0.022	0.863 ± 0.002	0.883 ± 0.056	0.836 ± 0.082	0.862 ± 0.091
RLAS-BSSA	0.747 ± 0.029	0.769 ± 0.072	0.750 ± 0.042	0.763 ± 0.025	0.746 ± 0.041	0.755 ± 0.001
RLAS-BCOA	0.860 ± 0.085	0.889 ± 0.089	0.882 ± 0.063	0.897 ± 0.237	0.872 ± 0.093	0.862 ± 0.017
RLAS-BHMS	0.849 ± 0.002	0.880 ± 0.123	0.879 ± 0.090	0.893 ± 0.036	0.840 ± 0.100	0.870 ± 0.009
RLAS-BGDM	0.752 ± 0.012	0.753 ± 0.027	0.769 ± 0.058	0.789 ± 0.085	0.731 ± 0.000	0.760 ± 0.018
RLAS-BABC	0.875 ± 0.004	0.906 ± 0.021	0.888 ± 0.046	0.900 ± 0.082	0.878 ± 0.016	0.889 ± 0.023

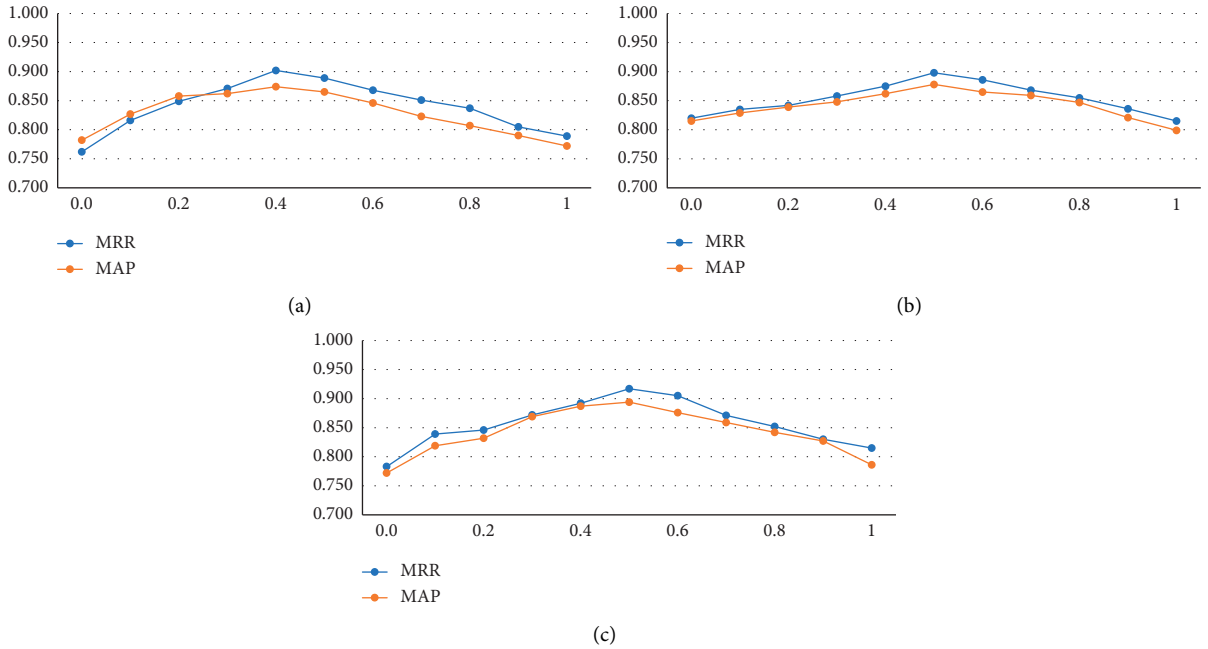


Figure 4: The process of changing the criteria by modifying the value of λ for the three datasets: (a) LegalQA dataset; (b) TrecQA dataset; (c) WikiQA dataset.

Table 6: The results of various loss functions on the model.

Model	Dataset					
	LegalQA		TrecQA		WikiQA	
	MAP	MRR	MAP	MRR	MAP	MRR
AS-BIABC + WCE	0.781 ± 0.002	0.819 ± 0.026	0.772 ± 0.005	0.780 ± 0.145	0.795 ± 0.010	0.792 ± 0.012
AS-BIABC + BCE	0.789 ± 0.000	0.812 ± 0.120	0.786 ± 0.073	0.804 ± 0.025	0.783 ± 0.074	0.814 ± 0.002
AS-BIABC + FL	0.842 ± 0.048	0.838 ± 0.056	0.839 ± 0.090	0.829 ± 0.012	0.832 ± 0.005	0.822 ± 0.006
AS-BIABC + DL	0.838 ± 0.089	0.808 ± 0.135	0.810 ± 0.074	0.770 ± 0.203	0.806 ± 0.082	0.804 ± 0.120
AS-BIABC + TL	0.785 ± 0.096	0.783 ± 0.582	0.821 ± 0.006	0.800 ± 0.041	0.823 ± 0.018	0.799 ± 0.005

Table 7: For the question "When were the Nobel Prize awards first given?" the table shows the top-5 answers from the model with and without reinforcement learning.

Rank	Ranked answers w/o RL	Ranked answers by RL
1	The first *awards* ceremony took place **in 1901**	The *award* to Doctors Without Borders echoes the first Nobel Peace Prize of the century, *given* **in 1901**, of which the founder of the red cross was a corecipient
2	The five-member *awards* committee works in secrecy during its five or six meetings a year and refuses to comment on or release candidates' names	The *prizes*, first awarded **in 1901**, are always presented on Dec 10, anniversary of *Nobel*'s death
3	**In 1901**, Sweden bestowed the inaugural Nobel *Prize* in Medicine on a Berliner, Emil von Behring, for his serum against diphtheria	Among them is the winner of the first *prize* **in 1901**, Sully Prudhomme
4	The *prizes*, first awarded **in 1901**, are always presented on Dec 10, anniversary of Nobel's death	The first *awards* ceremony took place **in 1901**
5	"We all know that there are still major problems to be faced," said *awards* committee chairman Francis Sejersted	A day after the announcement, for example, critic Norman Holmes Pearson grumbled that this woman, Pearl Buck, was *given* the Nobel *Prize* in Literature

"In 1901" is the ground truth answer, and italicized words are terms that appear in the question.

Table 8: The results of various word embeddings on the model.

Word embedding	Dataset					
	LegalQA		TrecQA		WikiQA	
	MAP	MRR	MAP	MRR	MAP	MRR
One-Hot encoding	0.679 ± 0.042	0.569 ± 0.002	0.711 ± 0.120	0.653 ± 0.081	0.649 ± 0.089	0.589 ± 0.093
CBOW	0.869 ± 0.006	0.843 ± 0.000	0.889 ± 0.078	0.869 ± 0.120	0.836 ± 0.012	0.828 ± 0.010
Skip-gram	0.874 ± 0.052	0.872 ± 0.075	0.878 ± 0.030	0.858 ± 0.002	0.847 ± 0.014	0.853 ± 0.014
GloVe	0.812 ± 0.027	0.853 ± 0.082	0.795 ± 0.140	0.821 ± 0.074	0.782 ± 0.039	0.806 ± 0.009
FastText	0.881 ± 0.002	0.901 ± 0.041	0.886 ± 0.093	0.876 ± 0.002	0.861 ± 0.099	0.870 ± 0.000

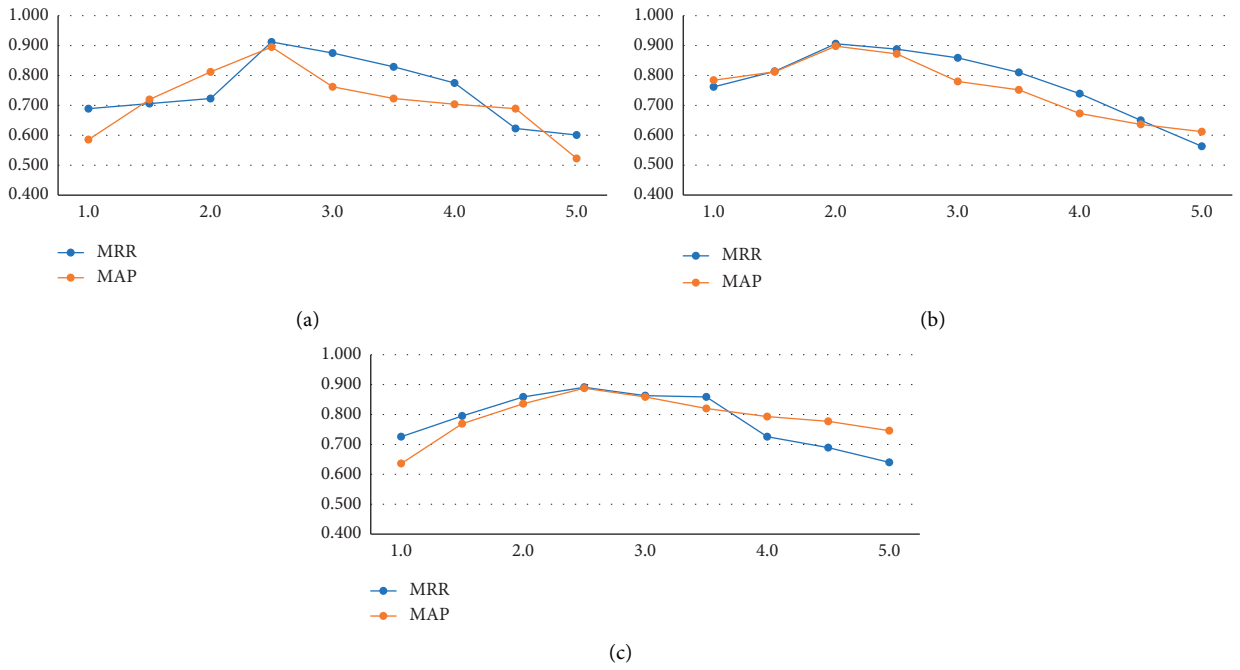


Figure 5: The process of changing the criteria by modifying the value of F for the three datasets: (a) LegalQA dataset; (b) TrecQA dataset; (c) WikiQA dataset.

5.5.4. Exploration on Word Embedding. Word embedding is one of the main components of deep learning models because the input is interpreted as a vector, and in case of incorrect embedding, the model may be misled. This study uses the BERT model as a word embedding, developed as one of the latest embedding models. In order to check other word embeddings on our model, we employ four word embeddings: One-Hot encoding [140], CBOW [141], Skip-gram [93], GloVe [94], and FastText [95]. One-Hot encoding is the vital process of altering the categorical data variables to be supplied to deep learning algorithms, improving predictions and classification accuracy. This word embedding makes a new binary feature for each class and allocates a value of 1 to the feature of each sample that corresponds to its original class. CBOW and Skip-gram are models that use neural networks to map a word to its embedding vector. The GloVe word embedding is an unsupervised learning algorithm performed on a corpus's aggregated global word-word cooccurrence statistics. FastText is word embedding that is an extension of the Skip-gram model. Instead of learning vectors for words, this method represents each word as an n-gram of characters. The results of this experiment are shown in Table 8. As expected, One-Hot encoding has the worst performance among all word embeddings, so in the TrecQA dataset, where this word embedding shows the best performance, the improvement rates for the MAP and MRR criteria are about 64.70% and 72.91%, respectively. CBOW and Skip-gram perform almost identically in all three datasets due to their similar architecture, with both being superior to the GloVe word embedding. FastText serves as the best word embedding for all models but still acts poorly on the BERT model. The BERT model decreases errors by more than 11%, 10%, and 19% compared to the FastText model for the WikiQA, TrecQA, and LegalQA datasets.

5.5.5. The Effect of the Parameter **F** *on the Model.* To examine the effect of the parameter F expressed by (10) on the proposed method algorithm performance, F is set to 0.5, 1, 1.5, 2, 2.5, 3.5, 4, 4.5, and 5. The results obtained by these settings for the three datasets are shown in Figure 5. As can be seen, for the LegalQA dataset, when F rises from 0 to 2,

the algorithm performs better and better. However, it can be observed that when F increases from 2 to 5, the method performance decreases. This means that a small or large value of F weakens the algorithm performance. For the TrecQA and WikiQA datasets, the algorithm with F equal to 1.5 and 2 has the best performance compared to other values.

6. Conclusion and Future Works

This paper presented an approach called RLAS-BIABC for AS, established on an attention mechanism-based LSTM method and the BERT word embedding, combined with an improved ABC algorithm for pretraining and reinforcement learning for training the BP algorithm. The RLAS-AM-ABC model aims to classify the two positive and negative classes, in which the positive pair includes a question and a real answer. In contrast, the negative couple carries a question and a fake answer. Due to many negative pairs in the dataset, the RLAS-BIABC is converted to an imbalanced classification. To overcome this problem, we formulate our model as a sequential decision-making process. In this regard, the environment assigned a reward to each classification act at each step, where a minority class has a higher reward. It continued until the agent mistakenly categorized a minority class sample or the number of episodes ended. Initial weighting is another essential characteristic of deep models, which can result in getting stuck in a local optimum. To solve this concern, we initialized the policy weights with the improved ABC algorithm. The paper proposed a mutual learning technique that alters the produced candidate food source with the higher fitness between two individuals chosen by a mutual learning factor. We designed experiments to examine the factors influencing the model. The analyses demonstrate the power of reinforcement learning, BERT, and the improved ABC algorithm for selecting answers.

In future work, while improving the proposed model, we will try to examine the effectiveness of the proposed classifier on other NLP applications. Another task would be to provide a model for generating the answer to a question. As a solution, we will focus on GANs, which today has many applications in almost every field, including NLP tasks.

Data Availability

The data used to support the findings of this study are included within the article. We included the information of datasets in the articles (see part 5.1).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Y. Sun, C. Chen, T. Xia, and X. Zhao, "QuGAN: quasi generative adversarial network for Tibetan question answering corpus generation," *IEEE Access*, vol. 7, pp. 116247–116255, 2019.

[2] J. Huang, "A multi-size neural network with attention mechanism for answer selection," 2021, https://arxiv.org/abs/2105.03278.

[3] W. Lei, Y. Xiang, Y. Wang, Q. Zhong, M. Liu, and M.-Y. Kan, "Linguistic properties matter for implicit discourse relation recognition: combining semantic interaction, topic continuity and attribution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, p. 1, Louisiana, USA, February 2018.

[4] S. Li, W. Lei, Q. Wu, X. He, P. Jiang, and T.-S. Chua, "Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users," 2020, https://arxiv.org/abs/2005.12979.

[5] S. Zhang, X. Zhang, H. Wang, J. Cheng, P. Li, and Z. Ding, "Chinese medical question answer matching using end-to-end character-level multi-scale CNNs," *Applied Sciences*, vol. 7, no. 8, p. 767, 2017.

[6] H. Yu, M. Lee, D. Kaufman et al., "Development, implementation, and a cognitive evaluation of a definitional question answering system for physicians," *Journal of Biomedical Informatics*, vol. 40, no. 3, pp. 236–251, 2007.

[7] X. Xu, L. He, A. Shimada, R.-i. Taniguchi, and H. Lu, "Learning unified binary codes for cross-modal retrieval via latent semantic hashing," *Neurocomputing*, vol. 213, pp. 191–203, 2016.

[8] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2494–2507, 2017.

[9] S. Jain and T. Dodiya, "Rule based architecture for medical question answering system," in *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012)*, pp. 1225–1233, Springer, Jaipur, India, December, 2012.

[10] X. Yao, B. Van Durme, C. Callison-Burch, and P. Clark, "Answer extraction as sequence tagging with tree edit distance," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 858–867, Atlanta, Georgia, June, 2013.

[11] S.-J. Yen, Y.-C. Wu, J.-C. Yang, Y.-S. Lee, C.-J. Lee, and J.-J. Liu, "A support vector machine-based context-ranking model for question answering," *Information Sciences*, vol. 224, pp. 77–87, 2013.

[12] B. L. Cairns, R. D. Nielsen, J. J. Masanz et al., "The MiPACQ clinical question answering system," in *Proceedings of the AMIA... Annual Symposium proceedings. AMIA Symposium*, vol. 2011, pp. 171–80, American Medical Informatics Association, Washington, DC, USA, October 2011.

[13] W. Lu, H. Wu, P. Jian, Y. Huang, and H. Huang, "An empirical study of classifier combination based word sense disambiguation," *IEICE - Transactions on Info and Systems*, vol. E101.D, no. 1, pp. 225–233, 2018.

[14] S. Wang and L. Cao, "Inferring implicit rules by learning explicit and hidden item dependency," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 3, pp. 935–946, 2017.

[15] M. Heilman and N. A. Smith, "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," in *Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1011–1019, downtown Los Angeles, June 2010.

[16] H. Toba, Z.-Y. Ming, M. Adriani, and T.-S. Chua, "Discovering high quality answers in community question answering archives using a hierarchy of classifiers," *Information Sciences*, vol. 261, pp. 101–115, 2014.

[17] S. Moravvej, M. Maleki Kahaki, M. Salimi Sartakhti, and M. Joodaki, "Efficient GAN-based method for extractive summarization," *Journal of Electrical and Computer Engineering Innovations*, 2021.

[18] S. V. Moravvej, A. Mirzaei, and M. Safayani, "Biomedical text summarization using conditional generative adversarial network (CGAN)," 2021, https://arxiv.org/abs/2110.11870.

[19] M. S. Sartakhti, M. J. M. Kahaki, S. V. Moravvej, M. javadi Joortani, and A. Bagheri, "Persian language model based on BiLSTM model on COVID-19 corpus," in *Proceedings of the 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 1–5, IEEE, Kashan, Iran, April 2021.

[20] Z. Sobhaninia, H. Danesh, R. Kafieh, J. J. Balaji, and V. Lakshminarayanan, "Determination of foveal avascular zone parameters using a new location-aware deep-learning method," in *Applications of Machine Learning*, vol. 2021, p. 1184311, International Society for Optics and Photonics, 2021.

[21] Z. SobhaniNia, N. Karimi, P. Khadivi, R. Roshandel, and S. Samavi, "Brain tumor classification using medial residual encoder layers," 2020, https://arxiv.org/abs/2011.00628.

[22] Z. Sobhaninia, S. Rafiei, A. Emami et al., "Fetal ultrasound image segmentation for measuring biometric parameters using multi-task deep learning," in *Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 6545–6548, IEEE, Berlin, Germany, July 2019.

[23] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: fine-grained image generation through asymmetric training," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2745–2754, Venice, Italy, October 2017.

[24] P. Kim, "Convolutional neural network," in *MATLAB Deep Learning*, pp. 121–147, Apress, Berkeley, CA, 2017.

[25] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[26] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[28] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[29] J. Mozafari, A. Fatemi, and M. A. Nematbakhsh, "BAS: an answer selection method using BERT language model," 2019, https://arxiv.org/abs/1911.01528.

[30] X. Xin Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[32] C. Blum and K. Socha, "Training feed-forward neural networks with ant colony optimization: an application to pattern classification," in *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, p. 6, IEEE, Rio de Janeiro, Brazil, November 2005.

[33] R. S. Sexton, B. Alidaee, R. E. Dorsey, and J. D. Johnson, "Global optimization for artificial neural networks: a tabu search application," *European Journal of Operational Research*, vol. 106, no. 2-3, pp. 570–584, 1998.

[34] R. S. Sexton, R. E. Dorsey, and J. D. Johnson, "Optimization of neural networks: a comparative analysis of the genetic algorithm and simulated annealing," *European Journal of Operational Research*, vol. 114, no. 3, pp. 589–601, 1999.

[35] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[36] F. Glover and M. Laguna, "Tabu search," in *Handbook of Combinatorial Optimization*, pp. 2093–2229, Springer, Heidelberg, Germany, 1998.

[37] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.

[38] S. Mirjalili, "Genetic algorithm, Studies in Computational Intelligence," in *Evolutionary Algorithms and Neural Networks*, pp. 43–55, Springer, Heidelberg, Germany, 2019.

[39] C. A. de Sousa, "An overview on weight initialization methods for feedforward neural networks," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 52–59, IEEE, Vancouver, BC, Canada, July 2016.

[40] A. Ranganathan, "The levenberg-marquardt algorithm," *Tutoral on LM algorithm*, vol. 11, no. 1, pp. 101–110, 2004.

[41] C. Ozturk and D. Karaboga, "Hybrid artificial bee colony algorithm for neural network training," in *Proceedings of the 2011 IEEE congress of Evolutionary Computation (CEC)*, pp. 84–88, IEEE, New Orleans, LA, USA, July 2011.

[42] S. V. Moravvej, S. J. Mousavirad, M. H. Moghadam, and M. Saadatmand, "An LSTM-Based Plagiarism Detection via Attention Mechanism and a Population-Based Approach for Pre-training Parameters with Imbalanced Classes," *Neural Information Processing*, Springer International Publishing, vol. 13110, , pp. 690–701, 2021.

[43] Z. Abdi Alkareem Alyasseri, O. A. Alomari, M. A. Al-Betar et al., "EEG channel selection using multiobjective cuckoo search for person identification as protection system in healthcare applications," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 5974634, 18 pages, 2022.

[44] S. Vakilian, S. V. Moravvej, and A. Fanian, "Using the cuckoo algorithm to optimizing the response time and energy consumption cost of fog nodes by considering collaboration in the fog layer," in *Proceedings of the 2021 5th International Conference on Internet of Things and Applications (IoT)*, pp. 1–5, IEEE, University of Isfahan, Iran, May 2021.

[45] S. Vakilian, S. V. Moravvej, and A. Fanian, "Using the artificial bee colony (ABC) algorithm in collaboration with the fog nodes in the Internet of Things three-layer architecture," in *Proceedings of the 2021 29th Iranian Conference on Electrical Engineering (ICEE)*, pp. 509–513, IEEE, Tehran, Iran, May 2021.

[46] S. J. Mousavirad, G. Schaefer, I. Korovin, D. Oliva, and Rde-Op, "RDE-OP: a region-based differential evolution algorithm incorporation opposition-based learning for optimising the learning process of multi-layer neural networks," in *Proceedings of the International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 407–420, Springer, Seville, Spain, April 2021.

[47] H. Han, W.-Y. Wang, B.-H. Mao, and Borderline-Smote, "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning," in *Proceedings of the International Conference on Intelligent Computing*, pp. 878–887, Springer, Hefei, China, August 2005.

[48] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: a case study involving information extraction,"

in *Proceedings of the workshop on learning from imbalanced datasets*, vol. 126, ICML United States, Washington, DC, August 2003.

[49] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 4368–4374, IEEE, Vancouver, BC, Canada, July 2016.

[50] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5375–5384, Las Vegas, Nevada, July 2016.

[51] Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen, "Deep learning for imbalanced multimedia data classification," in *Proceedings of the 2015 IEEE International Symposium on Multimedia (ISM)*, pp. 483–488, IEEE, Miami, FL, USA, December 2015.

[52] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3573–3587, 2017.

[53] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1367–1381, 2018.

[54] Q. Dong, S. Gong, and X. Zhu, "Class rectification hard mining for imbalanced deep learning," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1851–1860, Venice, Italy, October 2017.

[55] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Playing Atari with Deep Reinforcement Learning," 2013, https://arxiv.org/abs/1312.5602.

[56] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3389–3396, IEEE, Singapore, May 2017.

[57] X. Zhao, L. Zhang, L. Xia, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for list-wise recommendations," 2017, https://arxiv.org/abs/1801.00209.

[58] S. Wan, M. Dras, R. Dale, and C. Paris, "Using dependency-based features to take the'para-farce'out of paraphrase," in *Proceedings of the Australasian language technology workshop*, pp. 131–138, Sydney, Australia, November 2006.

[59] G. A. Miller and WordNet, *An Electronic Lexical Database*, MIT press, Cambridge, MA, 1998.

[60] V. Punyakanok, D. Roth, and W.-t. Yih, "Natural Language Inference via Dependency Tree Mapping: An Application to Question Answering," 2004, https://www.ideals.illinois.edu/handle/2142/11100.

[61] M. Wang and C. D. Manning, "Probabilistic tree-edit models with structured latent variables for textual entailment and question answering," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 1164–1172, Beijing, China, August 2010.

[62] S. W.-t. Yih, M.-W. Chang, C. Meek, and A. Pastusiak, "Question Answering Using Enhanced Lexical Semantic Models," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria's capital, August 2013.

[63] J. M. Giorgi and G. D. Bader, "Towards reliable named entity recognition in the biomedical domain," *Bioinformatics*, vol. 36, no. 1, pp. 280–286, 2020.

[64] A. Severyn and A. Moschitti, "Automatic feature engineering for answer selection and extraction," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 458–467, Seattle, Washington, USA, October 2013.

[65] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," 2016, https://arxiv.org/abs/1611.01747.

[66] S. Yoon, F. Dernoncourt, D. S. Kim, T. Bui, and K. Jung, "A compare-aggregate model with latent clustering for answer selection," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2093–2096, Beijing, China, November 2019.

[67] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 373–382, Santiago Chile, August 2015.

[68] D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2, pp. 707–712, Beijing, China, July 2015.

[69] J. Bromley, J. W. Bentz, L. Bottou et al., "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 04, pp. 669–688, 1993.

[70] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," 2014, https://arxiv.org/abs/1412.1632.

[71] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: a study and an open task," in *Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 813–820, IEEE, Scottsdale, Arizona, USA, December 2015.

[72] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1576–1586, Lisbon, Portugal, September 2015.

[73] J. Rao, H. He, and J. Lin, "Noise-contrastive estimation for answer selection with deep neural networks," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1913–1916, Indianapolis Indiana, USA, October 2016.

[74] H. T. Madabushi, M. Lee, and J. Barnden, "Integrating question classification and deep learning for improved answer selection," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3283–3294, Santa Fe, New-Mexico, August 2018.

[75] S. Kamath, B. Grau, and Y. Ma, "Predicting and integrating expected answer types into a simple recurrent neural network model for answer sentence selection," *Computación Y Sistemas*, vol. 23, no. 3, 2019.

[76] S. Kumar, K. Dixit, and K. Shah, "Interpreting text classifiers by learning context-sensitive influence of words," in *Proceedings of the First Workshop on Trustworthy Natural Language Processing*, pp. 55–67, Online, June 2021.

[77] T. Lai, T. Bui, and S. Li, "A review on deep learning techniques applied to answer selection," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2132–2144, Santa Fe, New-Mexico, August 2018.

[78] Y. Zhang and Y. Peng, "Research on answer selection based on LSTM," in *Proceedings of the 2018 International Conference on Asian Language Processing (IALP)*, pp. 357–361, IEEE, Bandung, Indonesia, November 2018.

[79] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "aNMM: ranking short answer texts with attention-based neural matching model," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 287–296, Indianapolis Indiana, USA, October 2016.

[80] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[81] G. Shen, Y. Yang, and Z.-H. Deng, "Inter-weighted alignment network for sentence pair modeling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1179–1189, Copenhagen, Denmark, September 2017.

[82] Y. Tay, L. A. Tuan, and S. C. Hui, "Co-stack residual affinity networks with multi-level attention refinement for matching text sequences," 2018, https://arxiv.org/abs/1810.02938.

[83] H. He, J. Wieting, K. Gimpel, J. Rao, and J. Lin, "UMD-TTIC-UW at SemEval-2016 Task 1: attention-based multi-perspective convolutional neural networks for textual similarity measurement," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1103–1108, San Diego, California, USA, June 2016.

[84] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[85] S. Abujar, A. K. M. Masum, S. M. H. Chowdhury, M. Hasan, and S. A. Hossain, "Bengali text generation using bi-directional RNN," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, IEEE, IIT, Kanpur, July 2019.

[86] A. K. M. Masum, S. Abujar, M. A. I. Talukder, A. S. A. Rabby, and S. A. Hossain, "Abstractive method of text summarization with sequence to sequence RNNs," pp. 1–5, IEEE.

[87] Y. Yang, J. Zhou, J. Ai et al., "Video captioning by adversarial LSTM," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5600–5611, 2018.

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

[89] A. Graves, "Generating sequences with recurrent neural networks," 2013, https://arxiv.org/abs/1308.0850.

[90] S. V. Moravvej, M. Joodaki, M. J. M. Kahaki, and M. S. Sartakhti, "A method based on an attention mechanism to measure the similarity of two sentences," in *Proceedings of the 2021 7th International Conference on Web Research (ICWR)*, pp. 238–242, IEEE, Tehran, Iran, May 2021.

[91] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks : The Official Journal of the International Neural Network Society*, vol. 18, no. 5-6, pp. 602–610, 2005.

[92] J.-Á. González, E. Segarra, F. García-Granada, E. Sanchis, and L. r. ı.-F. Hurtado, "Siamese hierarchical attention networks for extractive summarization," *Journal of Intelligent and Fuzzy Systems*, vol. 36, no. 5, pp. 4599–4607, 2019.

[93] C. McCormick, "Word2vec tutorial-the skip-gram model," 2016, http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model.

[94] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014.

[95] S. Thavareesan and S. Mahesan, "Sentiment lexicon expansion using Word2vec and fastText for sentiment prediction in Tamil texts," in *Proceedings of the 2020 Moratuwa Engineering Research Conference (MERCon)*, pp. 272–276, IEEE, Moratuwa, Sri Lanka, July 2020.

[96] H. S. Alatawi, A. M. Alhothali, and K. M. Moria, "Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT," *IEEE Access*, vol. 9, pp. 106363–106374, 2021.

[97] J. Devlin and M.-W. Chang, "Open sourcing BERT: state-of-the-art pre-training for natural language processing," *Google AI Blog*, vol. 2, 2018.

[98] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[99] S. A. H. Minoofam, A. Bastanfard, and M. R. Keyvanpour, "TRCLA: a transfer learning approach to reduce negative transfer for cellular learning automata," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[100] Z. Movahedi and A. Bastanfard, "Toward Competitive Multi-Agents in Polo Game Based on Reinforcement learning," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26773–26793, 2021.

[101] M. A. Wiering, H. Van Hasselt, A.-D. Pietersma, and L. Schomaker, "Reinforcement learning algorithms for solving classification problems," in *Proceedings of the 2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 91–96, IEEE, Paris, France, April 2011.

[102] J. Feng, M. Huang, L. Zhao, Y. Yang, and X. Zhu, "Reinforcement learning for relation classification from noisy data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, p. 1, New Orleans, Louisiana, USA, February 2018.

[103] T. Zhang, M. Huang, and L. Zhao, "Learning structured representation for text classification via reinforcement learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, February 2018.

[104] D. Liu and T. Jiang, "Deep reinforcement learning for surgical gesture segmentation and classification," *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018*, in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 247–255, Springer, September 2018.

[105] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement learning with visual attention for vehicle classification," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 4, pp. 356–367, 2016.

[106] J. Janisch, T. Pevný, and V. Lisý, "Classification with costly features using deep reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3959–3966, 2019.

[107] C. Martinez, G. Perrin, E. Ramasso, and M. Rombaut, "A deep reinforcement learning approach for early classification of time series," in *Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2030–2034, IEEE, Rome, Italy, September 2018.

[108] L. Abdi and S. Hashemi, "An ensemble pruning approach based on reinforcement learning in presence of multi-class imbalanced data," *Advances in Intelligent Systems and*

Computing, in *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, pp. 589–600, Springer, Roorkee, India, December 2013.

[109] A. K. Dixit and J. J. Sherrerd, *Optimization in Economic Theory*, Oxford University Press on Demand, Oxford, England, 1990.

[110] E. M. Voorhees, "The evaluation of question answering systems: lessons learned from the TREC QA track," in *Question Answering: Strategy and Resources Workshop Program*, p. 6, Citeseer, New Jersey, USA, 2002.

[111] W. Huang, J. Jiang, Q. Qu, and M. Yang, "AILA: A Question Answering System in the Legal Domain," in *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 5258–5260, Yokohama, July 2020.

[112] Y. Yang, W.-t. Yih, and C. Meek, "Wikiqa: a challenge dataset for open-domain question answering," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018, Lisbon, Portugal, September 2015.

[113] Y. Shen, Y. Deng, M. Yang et al., "Knowledge-aware attentive neural network for ranking question answer pairs," in *Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 901–904, Ann Arbor, Michigan, July 2018.

[114] S. V. Moravvej, M. J. M. Kahaki, M. S. Sartakhti, and A. Mirzaei, "A method based on attention mechanism using bidirectional long-short term memory (BLSTM) for question answering," in *Proceedings of the 2021 29th Iranian Conference on Electrical Engineering (ICEE)*, pp. 460–464, IEEE, Tehran, Iran, Islamic Republic of, May 2021.

[115] Y. Matsubara, T. Vu, and A. Moschitti, "Reranking for efficient transformer-based answer selection," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1577–1580, Xi'an, China, July 2020.

[116] S. Kim, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6586–6593, 2019.

[117] Y. Song, Q. V. Hu, and L. He, "P-CNN: e," *Knowledge-Based Systems*, vol. 169, pp. 67–79, 2019.

[118] G. Bao, Y. Wei, X. Sun, and H. Zhang, "Double attention recurrent convolution neural network for answer selection," *Royal Society Open Science*, vol. 7, no. 5, p. 191517, 2020.

[119] Q. Wang, W. Wu, Y. Qi, and Z. Xin, "Combination of active learning and self-paced learning for deep answer selection with bayesian neural network," in *Proceedings of the ECAI 2020*, pp. 1587–1594, IOS Press, Santiago de Compostela, Spain, July 2020.

[120] W. Huang, Q. Qu, and M. Yang, "Interactive knowledge-enhanced attention network for answer selection," *Neural Computing and Applications*, vol. 32, no. 15, pp. 11343–11359, 2020.

[121] V. V. Phansalkar and P. S. Sastry, "Analysis of the back-propagation algorithm with momentum," *IEEE Transactions on Neural Networks*, vol. 5, no. 3, pp. 505-506, 1994.

[122] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing Co, Boston, MA, 1996.

[123] C.-C. Yu and B.-D. Liu, "A backpropagation algorithm with adaptive learning rate and momentum coefficient," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 2, pp. 1218–1223, IEEE, Honolulu, HI, USA, May 2002.

[124] R. Battiti, "First- and second-order methods for learning: between steepest descent and Newton's method," *Neural Computation*, vol. 4, no. 2, pp. 141–166, 1992.

[125] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proceedings of the international conference on neural networks (ICNN'97)*, vol. 3, pp. 1930–1935, IEEE, Houston, TX, USA, June 1997.

[126] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[127] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Proceedings of the Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, pp. 65–74, Springer, Granada, Spain, May 2010.

[128] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing & Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[129] D. Bairathi and D. Gopalani, "Salp swarm algorithm (SSA) for training feed-forward neural networks," in *Soft Computing for Problem Solving*, pp. 521–534, Springer, Singapore, 2019.

[130] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the 2009 World congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, IEEE, Coimbatore, India, December 2009.

[131] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017.

[132] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[133] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[134] V. Pihur, S. Datta, and S. Datta, "Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach," *Bioinformatics*, vol. 23, no. 13, pp. 1607–1615, 2007.

[135] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1395–1403, NW Washington, DC, USA, December 2015.

[136] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, Venice, Italy, October 2017.

[137] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 240–248, Springer, Heidelberg, Germany, 2017.

[138] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," *Machine Learning in Medical Imaging*, in *Proceedings of the International Workshop on*

Machine Learning in Medical Imaging, pp. 379–387, Quebec, Canada, Springer, September 2017.

[139] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–7, IEEE, Via del Mar, Chile, October 2020.

[140] G. Hackeling, *Mastering Machine Learning with Scikit-Learn*, Packt Publishing, Birmingham, UK, 2014.

[141] S. Sonkar, A. E. Waters, and R. G. Baraniuk, "Attention word embedding," 2020, https://arxiv.org/abs/2006.00988.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6140796, 13 pages
https://doi.org/10.1155/2022/6140796

Research Article

Brent Crude Oil Price Forecast Utilizing Deep Neural Network Architectures

Amir Daneshvar,[1] Maryam Ebrahimi ,[1] Fariba Salahi,[2] Maryam Rahmaty,[3] and Mahdi Homayounfar[4]

[1]*Department of Information Technology Management, Electronic Branch, Islamic Azad University, Tehran, Iran*
[2]*Department of Industrial Management, Electronic Branch, Islamic Azad University, Tehran, Iran*
[3]*Department of Management, Chalous Branch, Islamic Azad University, Chalous, Iran*
[4]*Department of Industrial Management, Rasht Branch, Islamic Azad University, Rasht, Iran*

Correspondence should be addressed to Maryam Ebrahimi; ebrahimim@modares.ac.ir

Received 4 March 2022; Revised 4 April 2022; Accepted 17 April 2022; Published 5 May 2022

Academic Editor: Diego Oliva

Brent crude oil is considered as one of the most important sources of crude oil pricing in the worldwide market, and it is used to set the price of two-thirds of the traded crude oil supplies in the world. To predict the price of Brent crude oil, LSTM and Bi-LSTM methods are applied, which are the architecture of the recursive neural network. Initially, the database creates the appropriate data for the period January 2015 to March 2021 from Brent crude oil price signals and daily data from a financial market, and then, the modeling process is performed via the use of MATLAB software. Also, about 90% of the data are for training and the remaining for validation and comparison. Using LSTM and Bi-LSTM neural networks, the network architecture has been worked on, and by adding the number of layers and changing the solvers (SGDM, RMSProp, and Adam), the errors of different models are compared with each other. Nonlinear techniques of artificial neural networks and deep learning were used for modeling. Then, the network architecture was worked on and the model error rate was evaluated by comparing different layers and solvents such as SGDM, RMSProp, and Adam. The superiority of SGDM solvent over others was shown, and finally, it can be mentioned as the superior method of modeling of price forecasting in Brent crude oil field. The results show that the model with two layers of LSTM and SGDM solver has less error and better accuracy.

1. Introduction

Oil is a very important energy source whose international price fluctuations affect all aspects of the economy. The exchange rate is one of the important channels that shows how the international oil price shock is reflected in the real economy and financial markets. Understanding the characteristics and trends of oil price fluctuations provides the basis for a deep understanding of system mechanisms and the gradual trend in related research. Given the very complex characteristics of oil prices, it is very difficult to make accurate forecasts [1]. Men naturally seek to decipher the phenomena around the past in order to be able to predict the behavior of such phenomena and react to possible future events [2, 3]. It is more obvious in economic terms. The main feature of markets today is change, and the world is witnessing new developments and innovations in human societies. The highly dynamic nature and constant changes in the capital market have led researchers and economists to think about the best ways to predict the future and make the right decisions [4, 5].

In addition, models are used for analysis, better understanding, forecasting, systems development, etc., and basically in many financial issues such as investment and risk, the upper limit of the final quality of solving a problem is determined. Since price signals have a complex physical structure and it is very difficult to extract the laws and factors governing these signals, especially in the global scale, its effective parameters are not precisely defined and calculated, so analytical modeling are not appropriate. The purpose of

this study was to present a model (or models) in a financial field by which the future price of Brent crude oil can be predicted with appropriate accuracy (minimum error). This issue will be modeled through the deep learning method. Using artificial intelligence and machine learning tools, a nonlinear model (or models) of artificial neural networks will be presented for price prediction. The most popular and widely used type of recursive neural network (RNN) is long short-term memory (LSTM). LSTM networks are able to solve the two main problems that exist in RNN, namely the disappearance slope and the explosion slope. The key to solving these problems is the internal structure used in LSTM. Here, LSTM and bidirectional long-short term memory (Bi-LSTM) are applied for the phase of modeling. Meta-heuristic methods such as colonial competition algorithm (CCA) or methods such as cross-validation will be used to optimize network parameters. The research hypotheses are as follows:

(i) The price signal contains information in its content

(ii) The future price of time-series data is highly dependent on its past price

(iii) The future price of time-series data depends on external factors related to it

(iv) The performance of an expert system depends on the quality of its training

Additionally, the authors are looking for the way how financial market time-series data can be predicted through deep learning method.

In the following, in Section 2, the research background is examined. Section 3 describes the research method. In Section 4, the results are numbered and analyzed, and finally in Section 5, conclusions are made.

2. Literature Review

Behradmehr [6] used wavelet transform and neural networks for New York crude oil and Gulf of Mexico crude oil over a period of low volatility to present a model with a more accurate prediction of New York crude oil prices and Gulf of Mexico crude oil than other models. In this hybrid model, the wavelet transform was used to reduce the noise level of the data and then the oil price was predicted by artificial neural networks with smoothed data. The results of this study indicated that the elimination of noise improves the performance of oil price forecasting. In another study conducted by Pourkazemi and Asadi [7], oil prices were predicted and compared using artificial neural networks and the ARIMA econometric linear model, which showed less error in neural networks. In this study, by adding the OECD countries' reserves as an input, the forecasting error is reduced.

Investigating the price gap of Brent crude oil and diesel fuel using econometric methods, neural networks and wavelet transform were performed by Zolfaghari et al. [8]. The purpose of this study was to investigate the factors affecting the price gap and test the principle of symmetry between Brent crude oil prices and diesel fuel prices. Based on the results of linear and nonlinear models in this study, the principle of symmetry is accepted in short-term fluctuations in crude oil prices. This is not the case with long-term fluctuations. Brent oil price forecast in 2013 was done by an innovative and combined method that was meta-analyzed registered at Urmia University [9]. The results of this study showed that the accuracy of the meta-analysis method is much higher than other linear and nonlinear methods (fuzzy and neural) and has the least difference with real data. The study of the effect of oil price on Tehran Stock Exchange market stress using wavelet analysis was conducted by Jafari et al. [10]. It has been one-way relationship from the oil market to the stock market.

Neural network is one of the intelligent data mining techniques that has been used by researchers in different regions for the last 10 years. Predicting and analyzing stock market data play an important role in today's economy. The various algorithms used for prediction can be categorized into linear models (autoregressive (AR) model, moving average (MA) model, autoregressive integrated moving average (ARIMA) model) and nonlinear models (autoregressive conditional heteroskedasticity (ARCH), generalized autoregressive conditional heteroscedasticity (GARCH) model, neural network: multilayer perceptron (MLP), recursive neural networks (RNNs), long short-term memory (LSTM), and convolution neural network (CNN)) to predict a company's stock price based on historical prices. They used two different stock markets, the National Stock Exchange (NSE) of India and the New York Stock Exchange (NYSE). CNN has been found to perform better than other models. Compared with the ARIMA model, it has been observed that neural networks perform better than the existing linear model ARIMA [11, 12].

In a study conducted by Gupta and Pandey [13], the price of crude oil was predicted using frequent neural networks based on long short-term memory (LSTM). In this research, they have tried to use different types of models using different periods, revisions, and other adjustment methods, in which the result was very promising and has provided a reasonable logical forecast of crude oil prices in the near future. To increase accuracy and stability, Güleryüz and Özden [14] used LSTM and FBPr to predict future trends in Brent crude oil prices relative to previous prices and to compare two models built using data sets of 32-year weekly oil prices from June 1988 to June 2020. The model was determined to be the best fit. The data set was divided into two parts: training data and testing data, of which 25 years of 32 years have been selected as training data and the remaining 7 years as test data to confirm the accuracy of the forecast data. The coefficient of determination (R2) for LSTM and FBPr models was 0.92 and 0.89 in the training phase and 0.89 and 0.62 in the test phase, respectively. According to the results, the LSTM model has superior results for predicting oil price trends.

Salvi et al. [15] in a study examined the LSTM neural network and used it to predict the future trend of Brent oil prices based on the previous price of Brent oil. In this study, 4 types of errors have been calculated to check the accuracy of the model and errors. The mean absolute error (MAE) and

root-mean-square error (RMSE) were 1.1962 and 1.9164, respectively. In a study by Chen et al. [16], using the deep learning model, they depicted the unknown complex nonlinear properties of crude oil price movement and also proposed a new model for combining crude oil price forecasting based on the deep learning model. Using the model, the major movement of crude oil prices was analyzed and modeled. The performance of the model was evaluated using price data in WTI crude oil markets. Experimental results showed that the model improves the prediction accuracy.

Moitra et al. [17] attempted to use short-term memory neural network instead of convolutional neural network to predict crude oil price. The results were promising and showed more accurate forecasts for crude oil prices in the coming days, and a hybrid model was presented for forecasting crude oil price that used sophisticated network analysis and LSTM algorithms. The research results showed that the model is more accurate and has more robustness and reliability. Aziz et al. [18] used RNN-LSTM networks to predict crude oil prices based on historical data along with other technical analysis indicators. The developed model was trained and evaluated against accuracy matrices to evaluate the network's ability to provide improved accuracy in crude oil price forecasting compared with other strategies. The result obtained from the model showed the promising ability of the RNN-LSTM algorithm to predict the movement of crude oil prices.

Jammazi and Aloui [19] predicted the global price of crude oil using empirical evidence of wavelet analysis and neural network modeling. Yao and Wang [20] proposed a multistage forecasting method based on experimental modal analysis (EMA), LSTM, and GM (1, 1) model due to the problem of crude oil price forecasting. It offered daily, weekly, and monthly crude oil prices, the results of which showed that this model has high accuracy, especially in terms of series showing long-term effects with lower frequency, and GM model (1, 1) has a good performance with the trend of forecasting crude oil prices.

Sivalingam et al. [21] used a new learning algorithm called extreme learning machine (ELM), which had good learning ability and generalizability. The period used for the study was from January 1, 2000, to April 31, 2014. Since the price of gold is related to other commodities, five commodities, including the old gold price data, the silver price, the crude oil price, the S&P 500 index, and the foreign exchange rate, were considered as inputs. This study also compared the three models such as ELM, multilayer perceptron (MLP), and radial basis function (RBF), and the results showed that the ELM algorithm has a nearly 3% increase in efficiency compared with other neural networks. Lin [22] presented how to build a gold price forecast model to understand the future trend of gold prices, using old data and the stock price technical index formula, thereby five values of the gold technical index as an independent variable and the price of gold of the next day as a dependent variable were calculated. Three prediction models including backpropagation neural network (BPN), multiple regression (MR), and principal component regression (PCR) were applied, and the results showed that the BPN model has advantage.

Mensi et al. [23] analyzed the frequency-time analysis of gold and oil prices with stock markets and a wavelet-based approach. They examined the correlations between the five major emerging stock markets: Brazil, Russia, India, China, and South Africa (BRICS) and crude oil, and Brent and gold prices. The results using the wavelet analysis approach showed that the BRICS index is correlated with the price of crude oil at low frequencies (long horizons). On the other hand, no evidence of cooperation between the BRICS stock markets and the price of gold has been found. The implications of these results for the BRICS commodity portfolio showed that portfolio risk is affected by the interaction between stocks and oil markets. Arévalo et al. [24] presented a high-frequency trading strategy using deep neural networks (DNNs). In this study, the neural network predicts the next minute. This output is converted to get the average price of the next predicted minute.

Azadeh et al. [25] designed a model for oil price forecasting. They proposed a flexible algorithm based on artificial neural network (ANN) and fuzzy regression (FR) to meet the optimal long-term oil price forecast in complex environments with uncertainty. Oil chains, crude oil distillation capacity, non-OECD oil consumption, US refinery capacity, and surplus capacity have been cited as economic indicators included in this study. Analysis of variance (ANOVA) and Duncan's multiple range test (DMRT) were then used to test the accuracy of the predictions obtained from the ANN and FR models. The result of the study was that in terms of mean percentage error (MPE), ANN models were significantly higher than FR models. Spearman's correlation test was also used.

Safari and Davallou [26] applied a hybrid model to predict oil prices. They focused on oil price forecasting due to its effect on many economic and noneconomic factors. Since factors such as economic growth, political events, and psychological expectations affect oil prices, oil price forecasting is highly uncertain. The exponential smoothing model (ESM), autoregressive integrated moving average (ARIMA), and nonlinear autoregressive (NAR) network were combined in a state-space model framework to increase prediction accuracy. Linear and nonlinear patterns have been identified in economic and financial timelines such as crude oil prices. In the proposed hybrid model (PHM), the weight of the variable time for each model was determined by the Kalman filter. PHM was used in the monthly prices of OPEC crude oil and WTI crude oil prices. Numerical results showed a reduction in prediction error using PHM compared with its constituent models, the equal weight hybrid model (EWH), the genetic algorithm weight hybrid model (GWH), and the Zhang's hybrid model (ZHM).

Ding [27] proposed a new method to predict the price of crude oil using artificial neural networks. Huang and Wang [28] presented a global crude oil price forecast and accurate estimation based on coordination using a random wavelet neural network. Yang [29] examined gold prices from July 2013 to June 2018, which aimed to analyze the daily price of gold in dollars in the first half of July 2018 with the use of the

ARIMA model. In addition, the study used AC, PAC, AIC, and BIC to estimate the models, and the results suggested that ARIMA is the best model for predicting gold dollar prices.

Alameer et al. [30] proposed a new model to accurately predict monthly gold price fluctuations. In this model, a method called the whale optimization algorithm (WOA) algorithm was used as a trainer to neural network (NN) and the results were compared with other models such as NN, particle swarm optimization (PSO)-NN, genetic algorithm (GA)-NN, and grey wolf optimization (GWO)-NN. In addition, ARIMA models have been used as a criterion for evaluating the capacity of the proposed model. Experimental results showed the superiority of the WOA-NN hybrid model over other models and the proposed WOA-NN model improves the prediction accuracy obtained from the classic NN, PSO-NN, GA-NN, and GWO-NN models. ARIMA has reduced the average error. Kristjanpoller and Minutolo [31] attempted to answer the question: "Is it possible to improve the prediction of oil price fluctuations with the use of a hybrid model by combining financial variables?" The main conclusion was that the hybrid model increased the accuracy of 30% fluctuation prediction compared with previous models. The financial variables used in the model that improved the forecast were as follows: Euro/Dollar and Yen/Dollar exchange rates and DJIA and FTSE stock market indices.

Jafarzadeh Ghoushchi et al. [32] provided an extended approach to the diagnosis of tumour location in breast cancer using deep learning. This study develops a new machine learning approach based on modified deep learning (DL) to diagnose the tumour location in breast cancer. In this study, the data obtained from the databases (BCDRD01) are developed and resized and divided into data sets. A simple architecture is used for the first group of experiments, one of which utilizes a weighted function to counter the class imbalance. The results indicate that convolutional neural networks (CNNs) are an appropriate option for the separation of breast cancer lesions.

Hamdi et al. [33] investigated the relationship between oil price fluctuations and stock markets with wavelet analysis. Using quantitative regression analysis for recurring series and turbulent series during the period 2006 to 2017, the amount of fluctuations between oil prices and sectoral indicators in the Gulf Cooperation Council (GCC) countries, the United Arab Emirates, Saudi Arabia, Qatar, Oman, Kuwait and Bahrain, was examined. It was found that all sectors were dependent on oil price fluctuations. However, the banking and insurance sectors were not very sensitive to oil price fluctuations during the 10-, 25-, and 75-day periods. In addition, the relationship and the degree of interdependence between oil prices and stock returns of the sectors in the frequency domain were estimated (Table 1).

3. Methodology

One of the most widely used methods for modeling time series is the deep learning method. Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Deep learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance [36]. The research process is in accordance with Table 2.

3.1. Collecting and Creating a Database of Oil Prices and Forming a Price Vector. The basis of modeling of forecasting system is valid, accurate, and reliable data, because erroneous data destroy all modeling validity (no matter how powerful the model is). All data used in this study are extracted from investing.com. It should be noted that this site is the first rank of Google SEO and the first rank of Alexa in investing. The data extracted for crude oil are the Brent North Sea crude oil signal. Figure 1 shows the weekly price changes in Brent crude oil closing in the last 33 years.

As much as the data are used at a smaller sampling time (weekly, daily), high-frequency signals affect the price and the noise naturally increases and this affects the accuracy of the modeling. In other words, it has an adverse effect, but on the other hand, by increasing the number of data in smaller time intervals, more accurate information can be taught to the model.

3.2. Data Normalization. Normalization is the process of organizing data in a database efficiently. In other words, normalization is the way in which data are scaled. Each data set has properties that these large-value properties may have a much greater effect on the cost function than low-value properties. This problem will be solved by normalizing the properties so that their values are in the same range. The normalization operation causes all data to be in the range (1 and 0). Several statistical methods have been proposed to normalize the data. In this study, the min-max method has been used according to equation (1). Figure 2 shows the normalized signal of Brent crude oil daily price from January 2015 to March 2021:

$$x = \frac{x - x_{Min}}{x_{Max} - x_{Min}}. \quad (1)$$

3.3. Selecting the Appropriate Interval. Appropriate signal intervals should be selected for model training, validation, and testing that cover the entire signal dynamics to an acceptable level. The older the selected range, the farther it is from today's realities and naturally has adverse effects on the quality, accuracy, and precision of the model. According to the mentioned points, the selected period is a period of 6 years from January 2015 to March 2021.

Table 1: Comparison of research tools conducted in previous years in the field of price forecasting.

No.		Nonlinear time series	Linear time series	IO-nonlinear model	IO-linear model	Deep learning	MLP NN	Oil	Gold
1	Azadeh et al. [25]			*		*		*	
2	Li [34]			*			*		*
3	Kristjanpoller and Minutolo [31]			*			*		*
4	Lin [22]			*	*		*		*
5	Mensi et al. [23]	*	*					*	*
6	Arévalo et al. [24]						*	*	
7	Safari and Davallou [26]	*	*			*		*	
8	Chen et al. [16]					*			
9	Gupta and Pandey [13]					*			
10	Yang [29]		*						*
11	Ding [27]			*			*	*	
12	Huang and Wang [28]			*			*	*	
13	Hiransha et al. [11]					*			
14	Alameer et al. [30]	*	*				*		*
15	Salvi et al. [15]					*			
16	Güleryüz and Özden [14]					*			
17	Wu et al. [35]					*			
18	Moitra et al. [17]					*			

Table 2: Research process.

No.	Activity
1	Collecting and creating an oil price database and forming a price vector
2	Normalization of raw data
3	Preprocessing
4	Selecting the appropriate interval
5	Design and implementation of algorithms for price prediction process
6	Modeling using expert systems and machine learning
7	Identifying external signals affecting crude oil price output
8	Comparison of models

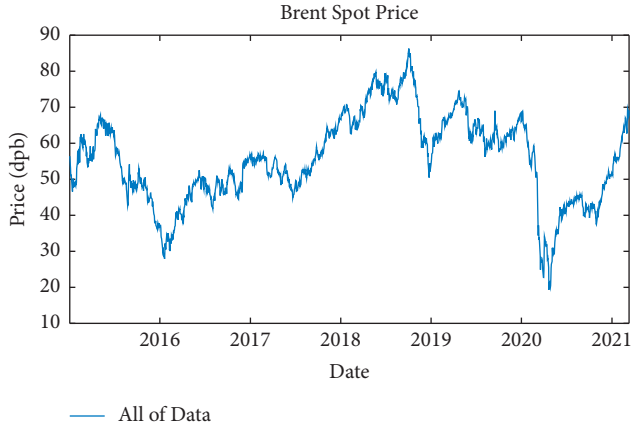


Figure 1: Weekly price changes in Brent crude oil closing in the last 33 years.

3.4. Design and Implementation of Algorithm for Price Forecasting Process. After data collection and database creation, the amount of training and test data ratios is determined experimentally. Raw data are then standardized and thermalized. Then, the network dynamics are designed to create the necessary structure for estimating the time series. In the next step, the short-term memory network architecture is designed and the network parameters are tuned. The network quality is evaluated, and finally, the network is updated to minimize errors.

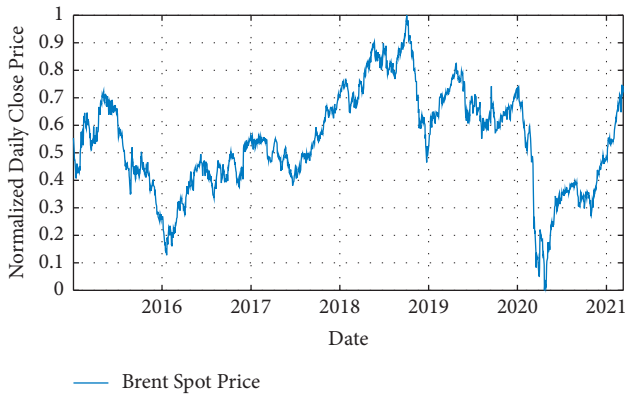


Figure 2: Normalized signal of Brent crude oil daily price from January 2015 to March 2021.

3.5. Modeling Using Expert Systems and Machine Learning. To predict the time series, the appropriate method for the model must be selected according to the signal behavior. There are several options for predicting a time series. Linear methods are usually the first choice in predicting time series. However, when the signal under study becomes complex and it is no longer possible to use linear methods, it is necessary to use a nonlinear method. Hence, various nonlinear methods have been developed to predict complex time series. Recursive neural tensor networks (RNTNs) including long short-term memory (LSTM) and bidirectional long-short term memory (Bi-LSTM) will be used to predict crude oil prices. The software used in this research is MATLAB R2018b.

4. Results

After processing the price signals and preparing them for use as a model and working on the network architecture, the proposed model with different layers and different solvents was examined. In one-layer LSTM with SGDM solver, 88% of the data is used as training data (from January 2015 to June 2020) and the remaining 12% as testing data (from June 2020 to March 2021); besides, in two-layer LSTM and three-layer LSTM with SGDM solver, 95% of the data is applied as training data (from January 2015 to January 2021) and the remaining 5% as testing data (from January 2021 to March 2021), which are shown in Figures 3–5.

In one-layer LSTM with RMSProp solver, 90% of the data is used as training data (from January 2015 to July 2020) and the remaining 10% as testing data (from July 2020 to March 2021), which is shown in Figure 6.

In one-layer LSTM with the Adam solver, 90% of the data is used as training data (from January 2015 to July 2020) and the remaining 10% as testing data (from July 2020 to March 2021), which is shown in Figure 7.

Initially, with only one-layer LSTM with SGDM solver, the root-mean-square error (RMSE) is equal to 1.88, maximum error is equal to 5.82, modeling time is equal to 429 seconds, mean squared error (MSE) is equal to 3.9, the number of feedback regressors is [1, 2, 3, 4], and the number of hidden layers is 100, which are shown in Table 3. Figures 8 and 9, respectively, provide comparison of system response

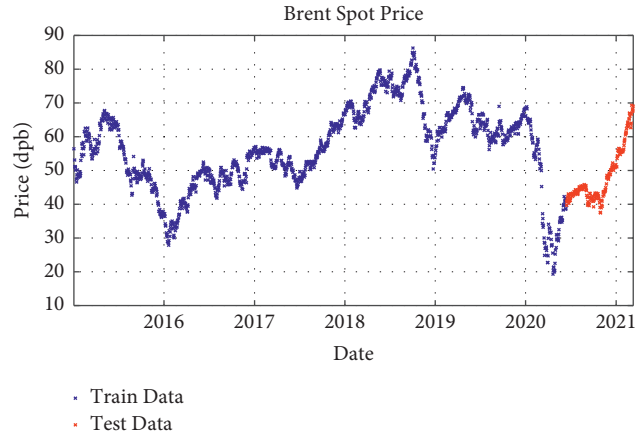


Figure 3: Training and test data for one-layer LSTM modeling with SGDM solver.

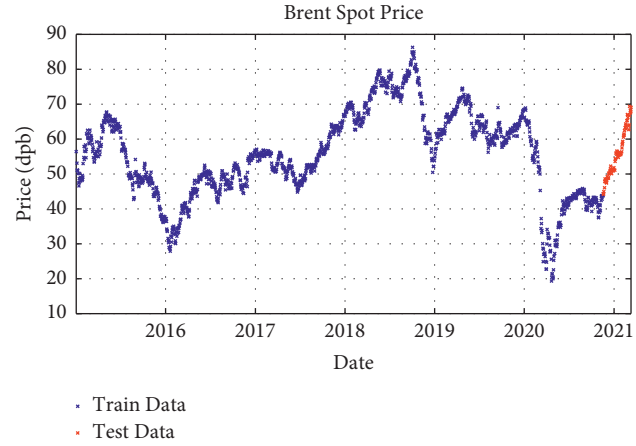


Figure 4: Training and test data for two-layer LSTM modeling with SGDM solver.

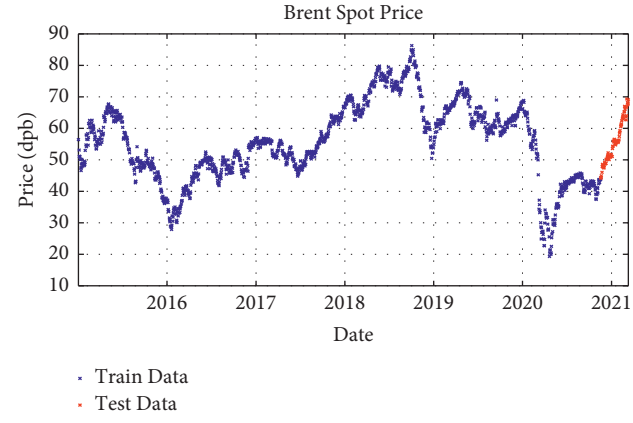


Figure 5: Training and test data for three-layer LSTM modeling with SGDM solver.

to real observations and comparison of system response to real observations (up-close) and their error.

Then, to improve the model error, the one-layer LSTM was changed to Bi-LSTM with SGDM solver. It is indicated that the RMSE is equal to 2.1838, maximum error is equal to

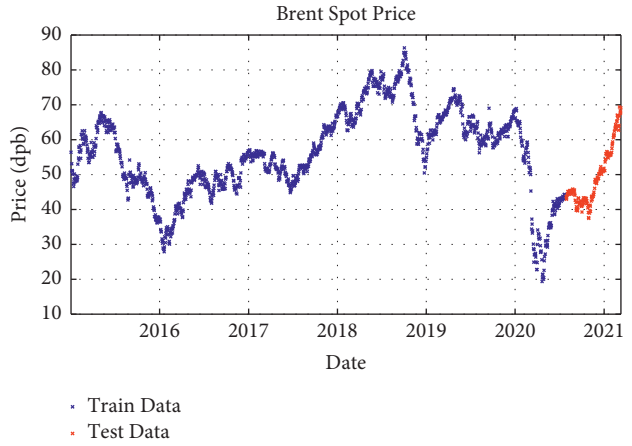


Figure 6: Training and test data for one-layer LSTM modeling with RMSProp solver.

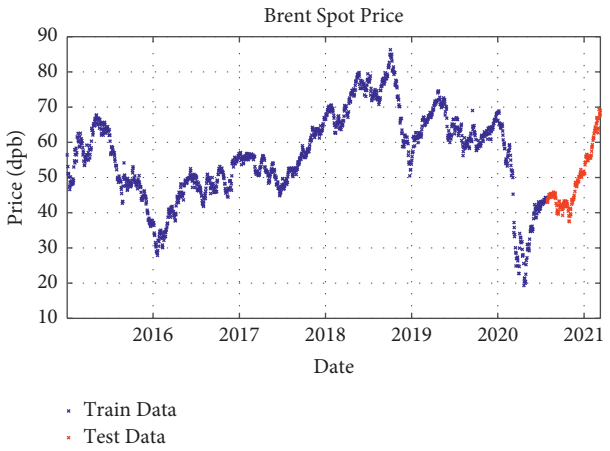


Figure 7: Training and test data for one-layer LSTM modeling with the Adam solver.

Table 3: Results of one-layer LSTM with SGDM solver for Brent crude oil.

Parameter	Value
Root-mean-square error (RMSE)	1.88
Maximum error	5.825
Modeling time	429
Mean squared error (MSE)	3.9
Feedback regressors	[1 2 3 4]
Hidden layers	100

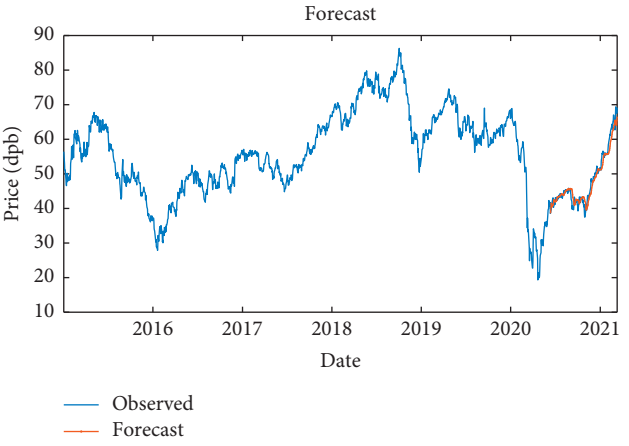


Figure 8: Comparison of system response to real observations of one-layer LSTM with SGDM solver.

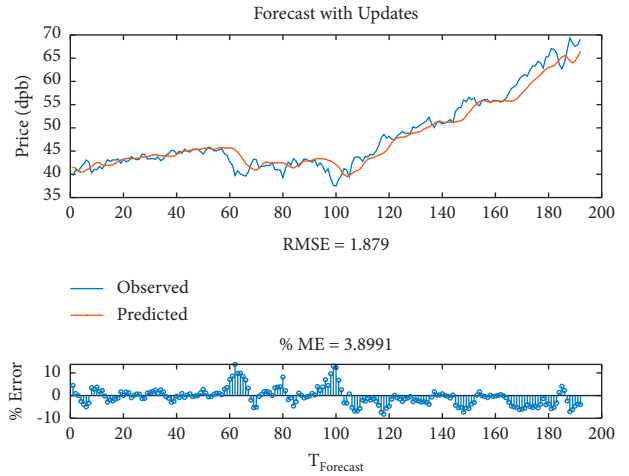


Figure 9: Comparison of system response to real observations (up-close) of one-layer LSTM with SGDM solver and their error.

Table 4: Results of Bi-LSTM with SGDM solver.

Parameter	Value
Root-mean-square error (RMSE)	2.18
Maximum error	7.74
Modeling time	822
Mean squared error (MSE)	3.60
Feedback regressors	[1 2 3 4]
Hidden layers	100

7.74, modeling time is equal to 822 seconds, MSE is equal to 3.60, the number of feedback regressors is [1, 2, 3, 4], and the number of hidden layers is 100, which are shown in Table 4. Figures 10 and 11, respectively, provide comparison of system response to real observations and comparison of system response to real observations (up-close) and their error.

By Bi-LSTM with SGDM solver, it is indicated that the RMSE is equal to 1.94, maximum error is equal to 11.24, modeling time is equal to 114 seconds, MSE is equal to 3.72, the number of feedback regressors is [1, 2], and the number of hidden layers is 10, which are shown in Table 5. Figures 12 and 13, respectively, provide comparison of system response to real observations and comparison of system response to real observations (up-close) and their error.

A Bi-LSTM layer with the Adam solver, like the previous case, did not have acceptable results for model evaluation and is not a suitable solvent.

Then, to improve the results, another LSTM layer was added to the previous layer, so that the model error might be corrected. It is shown that the RMSE is equal to 1.53, maximum error is equal to 7, modeling time is equal to

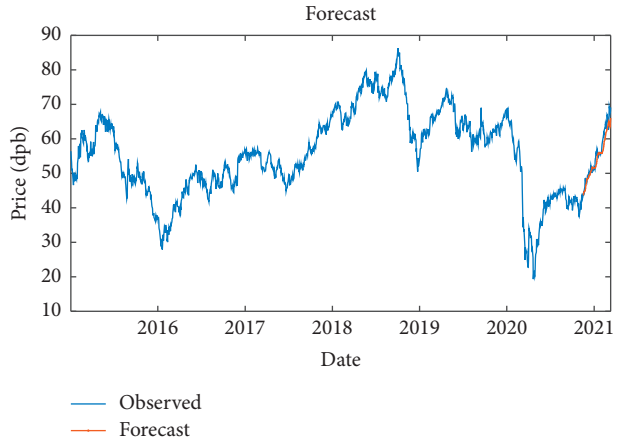


Figure 10: Comparison of system response to real observations of Bi-LSTM with SGDM solver.

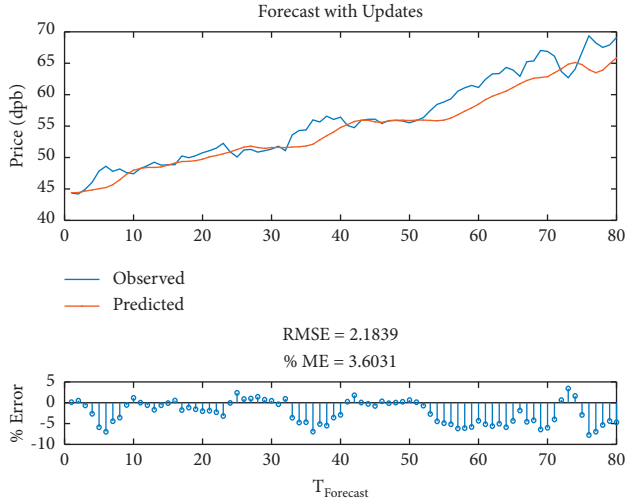


Figure 11: Comparison of system response to real observations (up-close) of Bi-LSTM with SGDM solver and their error.

Table 5: Results of Bi-LSTM with RMSProp solver.

Parameter	Value
Root-mean-square error (RMSE)	1.94
Maximum error	11.24
Modeling time	114
Mean squared error (MSE)	3.72
Feedback regressors	[1, 2]
Hidden layers	10

345.32 seconds, MSE is equal to 2.68, the number of feedback regressors is [1, 2], and the number of hidden layers is 80, which are shown in Table 6. Figures 14 and 15, respectively, provide comparison of system response to real observations and comparison of system response to real observations (up close) and their error.

Finally, the model with three-layer LSTM was tested with SGDM solver, whose RMSE is equal to 1.58, maximum error

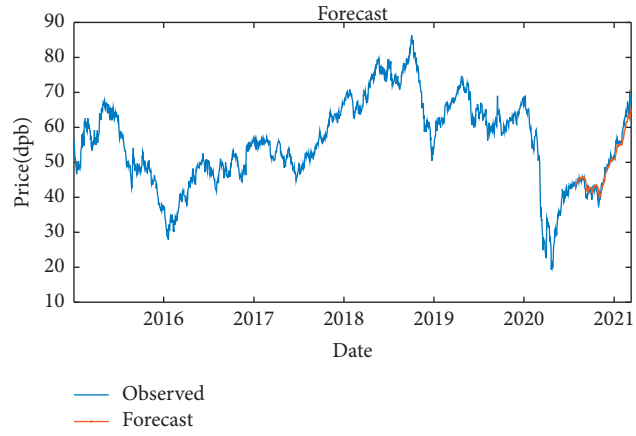


Figure 12: Comparison of system response to real observations of Bi-LSTM with RMSProp solver.

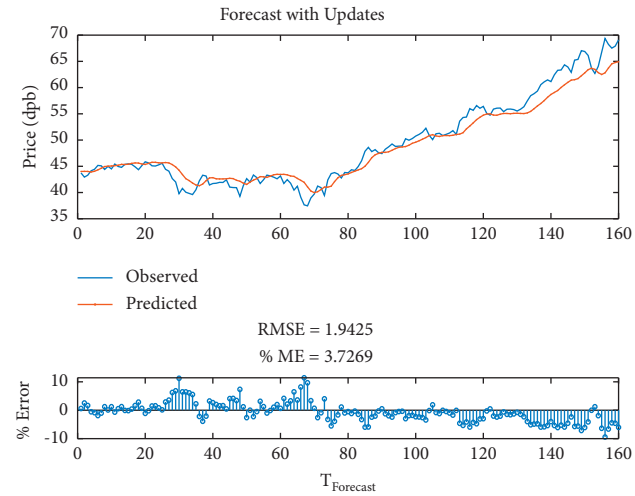


Figure 13: Comparison of system response to real observations (up-close) of Bi-LSTM with RMSProp solver and their error.

is equal to 5, modeling time is equal to 911 seconds, MSE is equal to 2.70, the number of feedback regressors is [1, 2, 3], and the number of hidden layers is 80, which are shown in Table 7. Figures 16 and 17, respectively, provide comparison of system response to real observations and comparison of system response to real observations (up-close) and their error.

At the end and in Table 8, all the models are compared with each other and they can be checked with the parameters. Two-layer LSTM with SGDM solver has less RMSE and MSE, and as a result the accuracy of the model is higher and better, but in terms of modeling time, one-layer LSTM with RMSProp solver is a more suitable option. To compare the number of hidden layers, one-layer LSTM with RMSProp solvent and a bi-LSTM layer with RMSProp solvent have fewer hidden layers, which again makes one-layer LSTM a better option than the Bi-LSTM layer.

Table 6: Results of two-layer LSTM with SGDM solver.

Parameter	Value
Root-mean-square error (RMSE)	1.53
Maximum error	7
Modeling time	345.32
Mean squared error (MSE)	2.68
Feedback regressors	[1, 2]
Hidden layers	80

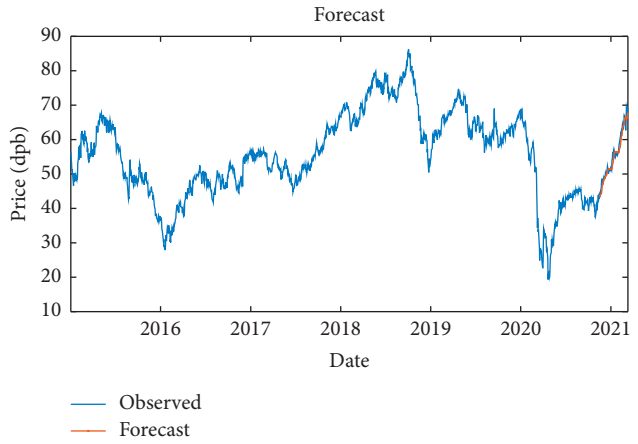


Figure 14: Comparison of system response to real observations of two-layer LSTM with SGDM solver.

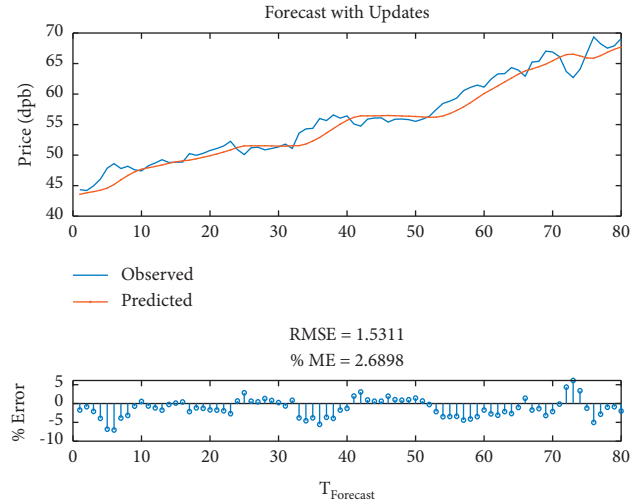


Figure 15: Comparison of system response to real observations (up-close) of two-layer LSTM with SGDM solver and their error.

Table 7: Results of three-layer LSTM with SGDM solver.

Parameter	Value
Root-mean-square error (RMSE)	1.58
Maximum error	5
Modeling time	911
Mean squared error (MSE)	2.70
Feedback regressors	[1, 2, 3]
Hidden layers	80

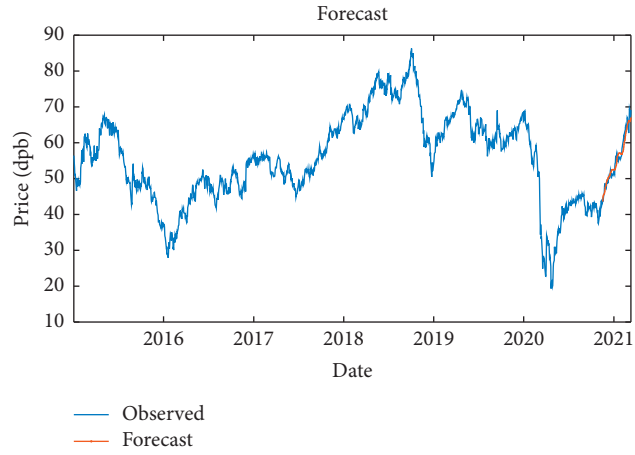


Figure 16: Comparison of system response to real observations of three-layer LSTM with SGDM solver.

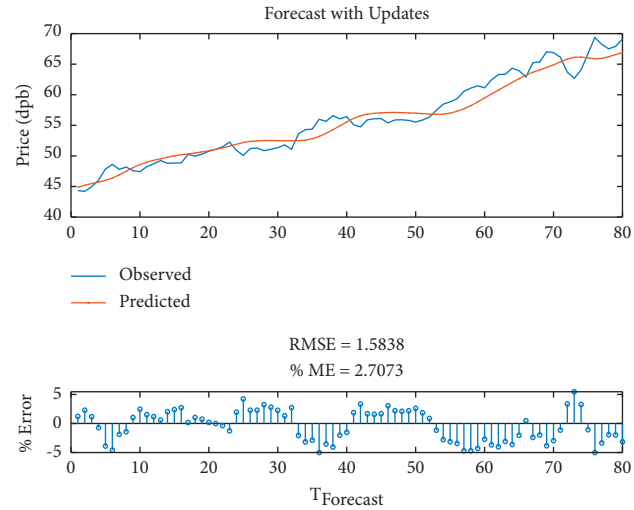


Figure 17: Comparison of system response to real observations (up-close) of three-layer LSTM with SGDM solver and their error.

Table 8: Comparison of modeling results.

Parameter	One-layer LSTM with SGDM solver	One-layer LSTM with RMSProp solver	One-layer LSTM with Adam solver	A Bi-LSTM with SGDM solver	A Bi-LSTM with RMSProp solver	Two-layer LSTM with SGDM solver	Three-layer LSTM with SGDM solver
RMSE	1.88	1.59	2.73	2.18	1.94	1.53	1.58
Maximum error	5.825	12.44	17.31	7.74	11.24	7	5
Modeling time	429	63.75	248	822	114	345.32	911
MSE	3.9	3.36	5.76	3.60	3.72	2.68	2.70
Feedback regressors	[1, 2, 3, 4]	[1, 2, 3, 4]	[1, 2, 7]	[1, 2, 3, 4]	[1, 2]	[1, 2]	[1, 2, 3]
Hidden layers	100	10	80	100	10	80	80

5. Conclusion

The main question of this research was as follows: "How can the price of time secret data be predicted by the deep learning method?" The answer was presented by collecting an appropriate database, reviewing and selecting modeling methods and efficient algorithms appropriate to the type of signal, and identifying as many effective factors and parameters as possible. It is indicated that the price of Brent crude oil in the global market can be predicted.

Then, in response to the question "How much does the future price of Brent crude oil depend on its past price?" different answers were given in different models, but by comparing the models, it is concluded that the price of crude

oil is dependent on the price of at least two days ago. The question then arose, "Does the future price of Brent crude oil depend on external factors related to it?" Examining the external signals, it was found that the price of crude oil is correlated with the factors and prices of other signals, such as gold and the Canadian dollar. Finally, in response to the question "Does the performance of an expert system depend on the quality of its training?" comparing the models, especially the neural network models with the recurrent and feed-forward training structures, it can be concluded that the performance of expert systems in the field of machine learning is highly dependent on the type and quality of their training.

In this study, first by collecting daily data from a reliable source, a database was formed to model and predict the price of Brent crude oil signals. The data period for modeling was selected from January 2015 to March 2021.

In this study, deep learning method was used to model and predict the price of Brent crude oil price signals. The reason for choosing this method was the complex dynamics of price signals and the lack of accurate information. Nonlinear techniques of artificial neural networks and deep learning were used for modeling. Then, the network architecture was worked on and the model error rate was evaluated by comparing different layers and solvents such as SGDM, RMSProp, and Adam. The superiority of SGDM solvent over others was shown, and finally, it can be mentioned as the superior method of modeling of price forecasting in Brent crude oil field.

The following points are summarized from this research:

(i) Crude oil price signals exhibit highly nonlinear and complex behavior

(ii) The SGDM solvent has less error in predicting Brent crude oil price for the designed model than RMSProp and Adam solvent

(iii) The RMSProp solvent also has less error than the Adam solvent in predicting Brent crude oil price for the designed model

(iv) By adding an LSTM layer to the network structure, the error model was improved, resulting in a more accurate result

(v) LSTM layers perform better than Bi-LSTM layers and further reduce model error

5.1. Contributions

(i) In this study, unlike all previous studies on Brent crude oil forecasting, the model network architecture has been worked on and an attempt has been made to improve the model error by changing layers and solvents. Other research differs from that conducted in this study. For example, in a study by Chen et al. [16], using a deep learning model, complex nonlinear features of unknown crude oil price movements were depicted. Also, a new model is proposed for crude oil price forecasting based on the deep learning model, which differs from the method used in this study.

(ii) Moitra et al. [17], in a study entitled "Crude oil price prediction using the LSTM method," have tried to use short-term memory neural network instead of convolutional neural network to predict crude oil price. They have used complex network analysis and LSTM and deep learning algorithms, while in this study we authors tried to provide a better and more appropriate model by changing the LSTM layer to the Bi-LSTM layer.

(iii) Furthermore, in existing studies no comparison was made to present and select the best model in the field of Brent crude oil price forecasting.

Brokers, traders, and investment advisers working in the field of energy, especially oil, are advised to predict the price of oil and take risk hedging measures to actively manage their portfolio using this model.

To improve the quality of this research, suggestions for future studies are presented, which are as follows:

(i) Designing a crude oil price forecasting model with other neural networks

(ii) Using other data processing methods such as STFT on crude oil signal and comparing results

Constraints are an integral part of research. However, in general, the limitations of research included the following:

(i) The strong dependence of the results on the type of modeling

(ii) Coronavirus pandemic during the research period

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] K. Czech and I. Niftiyev, "The impact of oil price shocks on oil-dependent countries' currencies: the case of Azerbaijan and Kazakhstan," *Journal of Risk and Financial Management*, vol. 14, no. 9, p. 431, 2021.

[2] A. Ghadiri Moghadam, "Studying the abilities of forecast models of Altman and Ahleson bankruptcy for bankruptcy forecast of accepted corporate in Stock Exchange market," *Knowledge and Development Magazine*, vol. 16, no. 28, pp. 193–220, 2009.

[3] A. Gaspar, D. Oliva, S. Hinojosa, I. Aranguren, and D. Zaldivar, "An optimized Kernel Extreme Learning Machine for the classification of the autism spectrum disorder by using gaze tracking images," *Applied Soft Computing*, vol. 120, 2022.

[4] H. Moradi, I. Jokar, and A. Forouzantabar, "Modeling and forecasting gold price using GMDH neural network," *Indian*

Journal of Fundamental and Applied Life Sciences, vol. 5, no. S1, pp. 30–41, 2015.

[5] H. Ayaz, E. Rodríguez-Esparza, M. Ahmad, D. Oliva, M. Pérez-Cisneros, and R. Sarkar, "Classification of apple disease based on non-linear deep features," *Applied Sciences*, vol. 11, no. 14, p. 6422, 2021.

[6] N. Behradmehr, "Crude oil price forecasting using wavelet transform and artificial neural network," *Quarterly Journal of Energy Economics Studies*, vol. 5, no. 18, pp. 81–98, 2008, [In Persian].

[7] M. H. Pourkazemi and M. B. Asadi, "Predicting the dynamic price of crude oil using artificial neural networks and using the oil reserves of OECD countries," *Journal of Economic Research*, vol. 44, no. 3, pp. 25–46, 2009, [In Persian].

[8] M. Zolfaghari, M. Najjar Firoozjaei, and B. Eryan, "Investigating the price gap of Brent crude oil and Diesel fuel using econometric methods, neural networks and wavelet transform," *Economic Modeling Research Quarterly*, vol. 4, no. 14, pp. 25–58, 2013.

[9] K. Shahbazi and S. Soleimian, "Oil price forecasting using meta-analysis method," *Quarterly Journal of Energy Economics Studies*, vol. 11, no. 47, pp. 67–93, 2015, [In Persian].

[10] M. Jafari, S. Fattahi, and K. Soheili, "Investigating the effect of oil price on Tehran Stock Exchange market stress using wavelet analysis. National Conference on Economics, Development Management and Entrepreneurship with the approach of supporting Iranian goods," *Zahedan, Industrial Management Organization of Sistan and Baluchestan*, vol. 17, no. 2, 2017.

[11] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "NSE stock market prediction using deep-learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018.

[12] S. Ganguly, P. Bhowal, D. Oliva, and R. Sarkar, "BLeafNet: a Bonferroni mean operator based fusion of CNN models for plant identification using leaf image classification," *Ecological Informatics*, vol. 2022, Article ID 101585, 2022.

[13] V. Gupta and A. Pandey, "Crude oil price prediction using LSTM networks," *International Journal of Computer and Information Engineering*, vol. 12, no. 3, pp. 226–230, 2018.

[14] D. Güleryüz and E. Özden, "The prediction of Brent crude oil trend using LSTM and Facebook prophet," *Avrupa Bilim ve Teknoloji Dergisi*, vol. 20, pp. 1–9, 2020.

[15] H. Salvi, A. Shah, M. Mehta, and S. Correia, "Long short-term model for Brent oil price forecasting," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, pp. 315–319, 2019.

[16] Y. Chen, K. He, and G. K. Tso, "Forecasting crude oil prices: a deep learning," *Procedia Computer Science*, vol. 122, pp. 300–307, 2017.

[17] N. Moitra, P. Raj, S. Saxena, and R. Kumar, "Crude oil prediction using LSTM," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 2, pp. 900–905, 2020.

[18] N. Aziz, M. H. A. Abdullah, and A. N. Zaidi, "Predictive analytics for crude oil price using RNN-LSTM neural network," in *Proceedings of the 2020 International Conference on Computational Intelligence (ICCI)*, pp. 173–178, IEEE, Bandar Seri Iskandar, Malaysia, 09 November 2020.

[19] R. Jammazi and C. Aloui, "Wavelet decomposition and regime shifts: assessing the effects of crude oil shocks on stock market returns," *Energy Policy*, vol. 38, no. 3, pp. 1415–1435, 2010.

[20] T. Yao and Z. Wang, "Crude oil price prediction based on LSTM network and GM (1, 1) model," *Grey Systems: Theory and Application*, vol. 11, no. 1, pp. 80–94, 2020.

[21] K. C. Sivalingam, S. Mahendran, and S. Natarajan, "Forecasting gold prices based on extreme learning machine," *International Journal of Computers, Communications & Control*, vol. 11, no. 3, pp. 372–380, 2016.

[22] C. Lin, "Build prediction models for gold prices based on back-propagation neural network," in *Proceedings of the International Conference on Modeling, Simulation and Applied Mathematics*, Advances in Intelligent Systems Research Atlantis Press, August 2015.

[23] W. Mensi, B. Hkiri, K. H. Al-Yahyaee, and S. H. Kang, "Analyzing time–frequency co-movements across gold and oil prices with BRICS stock markets: a VaR based on wavelet approach," *International Review of Economics & Finance*, vol. 54, pp. 74–102, 2018.

[24] A. Arévalo, J. Niño, G. Hernandez, J. Sandoval, and D. León, "Algorithmic trading using deep neural networks on high frequency data," in *Proceedings of the Workshop on Engineering Applications*, pp. 144–155, Springer, Cartagena, Colombia, September 2017.

[25] A. Azadeh, M. Moghaddam, M. Khakzad, and V. Ebrahimipour, "A flexible neural network-fuzzy mathematical programming algorithm for improvement of oil price estimation and forecasting," *Computers & Industrial Engineering*, vol. 62, no. 2, pp. 421–430, 2012.

[26] A. Safari and M. Davallou, "Oil price forecasting using a hybrid model," *Energy*, vol. 148, pp. 49–58, 2018.

[27] Y. Ding, "A novel decompose-ensemble methodology with AIC-ANN approach for crude oil forecasting," *Energy*, vol. 154, pp. 328–336, 2018.

[28] L. Huang and J. Wang, "Global crude oil price prediction and synchronization based accuracy evaluation using random wavelet neural network," *Energy*, vol. 151, pp. 875–888, 2018.

[29] X. Yang, "The prediction of gold price using ARIMA model," in *Proceedings of the 2nd International Conference on Social Science, Public Health and Education (SSPHE 2018)*, Atlantis Press, January 2018.

[30] Z. Alameer, M. Abd Elaziz, A. A. Ewees, H. Ye, and Z. Jianhua, "Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm," *Resources Policy*, vol. 61, pp. 250–260, 2019.

[31] W. Kristjanpoller and M. C. Minutolo, "Forecasting volatility of oil price using an artificial neural network-GARCH model," *Expert Systems with Applications*, vol. 65, pp. 233–241, 2016.

[32] S. Jafarzadeh Ghoushchi, R. Ranjbarzadeh, S. A. Najafabadi et al.,et al "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[33] B. Hamdi, M. Aloui, F. Alqahtani, and A. Tiwari, "Relationship between the oil price volatility and sectoral stock markets in oil-exporting economies: evidence from wavelet nonlinear denoised based quantile and Granger-causality analysis," *Energy Economics*, vol. 80, pp. 536–552, 2019.

[34] B. Li, "Research on WNN modeling for gold price forecasting based on improved artificial bee colony algorithm," *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 270658, 2014.

[35] Y. X. Wu, Q. B. Wu, and J. Q. Zhu, "Improved EEMD-based crude oil price forecasting using LSTM networks," *Physica A: Statistical Mechanics and its Applications*, vol. 516, pp. 114–124, 2019.

[36] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3642–3649, Manno-Lugano, Switzerland, 2012.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 8014979, 15 pages
https://doi.org/10.1155/2022/8014979

Research Article

Diagnosis of Retinal Diseases Based on Bayesian Optimization Deep Learning Network Using Optical Coherence Tomography Images

Malliga Subramanian,[1] M. Sandeep Kumar,[2] V. E. Sathishkumar,[3] Jayagopal Prabhu ,[2] Alagar Karthick,[4] S. Sankar Ganesh ,[5] and Mahseena Akter Meem [6]

[1]*Department of Computer Science Engineering, Kongu Engineering College, Perundurai, Erode 638060, Tamil Nadu, India*
[2]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India*
[3]*Department of Industrial Engineering, Hanyang University, Seoul, Republic of Korea*
[4]*Renewable Energy Lab, Department of Electrical and Electronics Engineering, KPR Institute of Engineering and Technology, Coimbatore 641407, Tamil Nadu, India*
[5]*Department of Artificial Intelligence and Data Science, KPR Institute of Engineering and Technology, Coimbatore 641407, Tamil Nadu, India*
[6]*Department of Electrical and Electronic Engineering, Daffodil International University, Ashulia, Savar, Dhaka 1207, Bangladesh*

Correspondence should be addressed to Jayagopal Prabhu; jprabhuit@gmail.com

Received 10 February 2022; Accepted 17 March 2022; Published 15 April 2022

Academic Editor: Ripon Chakrabortty

Retinal abnormalities have emerged as a serious public health concern in recent years and can manifest gradually and without warning. These diseases can affect any part of the retina, causing vision impairment and indeed blindness in extreme cases. This necessitates the development of automated approaches to detect retinal diseases more precisely and, preferably, earlier. In this paper, we examine transfer learning of pretrained convolutional neural network (CNN) and then transfer it to detect retinal problems from Optical Coherence Tomography (OCT) images. In this study, pretrained CNN models, namely, VGG16, DenseNet201, InceptionV3, and Xception, are used to classify seven different retinal diseases from a dataset of images with and without retinal diseases. In addition, to choose optimum values for hyperparameters, Bayesian optimization is applied, and image augmentation is used to increase the generalization capabilities of the developed models. This research also provides a comparison of the proposed models as well as an analysis of them. The accuracy achieved using DenseNet201 on the Retinal OCT Image dataset is more than 99% and offers a good level of accuracy in classifying retinal diseases compared to other approaches, which only detect a small number of retinal diseases.

1. Introduction

Healthcare diagnosis is a primary focus area of deep learning research, with major industry players like GE Healthcare [1] investing heavily in it. Deep learning-based applications such as face recognition in phones, object recognition and detection, security systems, number plate detection, and a slew of other industrial applications have already gone commercial. These marketed applications are less vulnerable to errors and misclassifications than potential healthcare applications where mistakes might cost lives. As a result, accuracy in medical image analysis is critical in healthcare-based applications [2, 3], and focused research is required to make algorithms robust. In recent years, retinal diseases have become a severe public health concern. They develop slowly and without noticeable indications. Every year, millions of individuals throughout the world are diagnosed with retinal diseases, and these diseases express themselves in several ways. Retinal diseases may damage any area of the retina, causing vision impairments, and some can ultimately

lead to blindness. Various retinal diseases include diabetic retinopathy (DR), macular pucker, glaucoma, macular hole (MH), age-related macular degeneration (AMD), drusen, central serous retinopathy (CSR), macular edema, vitreous traction, and optic nerve anomalies. These ailments lead to (i) loss of vision, (ii) floaters and cobwebs, (iii) flashing lights, (iv) objects seeming smaller or larger than they are, (v) decrease in peripheral vision or presence of shadows, and (vi) distortion of straight lines.

The term "prevention of retinal disease" refers to steps performed in advance to lessen the probability of vision loss, as well as the degree and effect of vision loss. In around 80% of instances, blindness and visual impairment may be prevented. A modest precautionary step can have a tremendous impact. Ophthalmologists often diagnose and treat retinal diseases. An ophthalmologist performs a full eye examination and searches for abnormalities everywhere in the eye. The Amsler Grid Test, OCT, Indocyanine Green Angiography, Ultrasound, Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) are just a few of the procedures used to detect the location and severity of a disease. Among these, OCT is the most important screening tool for detecting rare retinal and optic nerve diseases, and three-dimensional retinal structural information is provided by OCT images using a light wave based approach [4]. Numerous researches have demonstrated that deep learning algorithms performed admirably when applied to medical image analysis for classification of skin diseases [5], cardiovascular diseases' risk prediction [6], lung cancer detection [7], and much more. These remarkable attempts encourage several studies to employ deep learning in diagnosing retinal diseases [8–11]. Since the introduction of deep learning techniques, OCT imaging has sparked a lot of interest in automated diagnosis for detecting a variety of retinal diseases [12]. But these studies were able to detect only a few types of retinal diseases such as Choroidal Neovascularization (CNV), Diabetic Macular Edema (DME), Drusen, DR, Glaucoma, AMD, CSR, MH [11, 13–20], and many more. Sample OCT images for a few retinal diseases and normal retina (without retinal disease) are presented in Figure 1.

Deep neural networks, notably the CNNs, are frequently employed in image classification tasks and have demonstrated substantial performance since 2012 [21–24]. CNN's study on medical image categorization has produced results that are comparable to those of human experts. CheXNet, for example, a 121-layer CNN that was trained on a dataset of over 100,000 frontal-view chest X-rays, outperformed the average performance of four radiologists. [25] provided a detailed overview of the uses of CNNs in medical image classification. Furthermore, previous attempts to classify and diagnose retinal diseases using OCT images have shown that standard deep architectures like VGGNet, DenseNet, and others may be ineffective because of their large parameter space. Transfer learning, on the other hand, might be a feasible strategy for dealing with enormous parameter spaces. Models can learn in one domain, where there is a lot of data, and then can transfer that knowledge to another domain, where there is not as much data. By leveraging previously trained models, we may train deep neural architectures that need a large number of learning parameters despite a low number of available images [26]. Deep learning models excel at learning from a large number of labeled cases [27], but they can only generalize to scenarios that were not seen during training. Overfitting and falling into a local optimum will occur when training samples are inadequate [28]. Furthermore, developing a deep learning model from scratch often needs a considerable amount of processing power and takes a long time. Transfer learning can help us in dealing with such scenarios. Another part of this research is that it applies Bayesian optimization to identify an ideal configuration for hyperparameters, as determining the best values for training CNN architectures is challenging. Hence, this work attempts to answer the following research questions:

RQ1: will the presently available datasets be enough to detect a variety of frequently occurring retinal diseases?

RQ2: how can pretrained CNN models be used to classify new datasets using transfer learning?

RQ3: how can we customize the pretrained models? Does customization of such models significantly improve the quality of classification?

RQ4: will the performance of the proposed models be improved by tuning hyperparameters appropriately?

The above research questions pave the way for classifying retinal diseases effectively. In this work, to address the above research questions, we developed a set of models using pretrained VGG16, DenseNet201, InceptionV3, and Xception architectures to automatically classify and detect retinal diseases from OCT images. To repurpose pretrained models, we use two strategies: freezing the convolution base (feature extractor) and training a few top convolution layers while freezing others (fine-tuner). Even though there have been more research attempts to identify retinal problems using CNN-based deep learning models, these studies have only tried to detect 3 to 5 types of commonly occurring retinal diseases from OCT images. In this work, we intend to build a few deep learning models that can detect seven different forms of retinal diseases, including AMD, CNV, DME, CSR, DR, Drusen, and MH. The developed models will categorize the OCT images as the normal or infected retina. We collected OCT images and divided them into eight categories to train and evaluate the models: seven for retinal diseases and one for normal. To the best of our knowledge, no work has addressed transfer learning with two fine-tuning processes combined with Bayesian Optimization for detecting a wide range of retinal diseases in the literature. The main contributions and novelty of this work can be summarized as follows:

(i) Providing an open-access dataset that contains OCT images (OCT Image Dataset) to help ophthalmologists in diagnosing diagnosis a wide range of retinal diseases by applying deep learning techniques

(ii) Transfer learning is used in a novel way as a feature extractor and fine-tuner to build a few classifiers

(iii) Tuning hyperparameters to find the optimal values using Bayesian Optimization (grid search, an

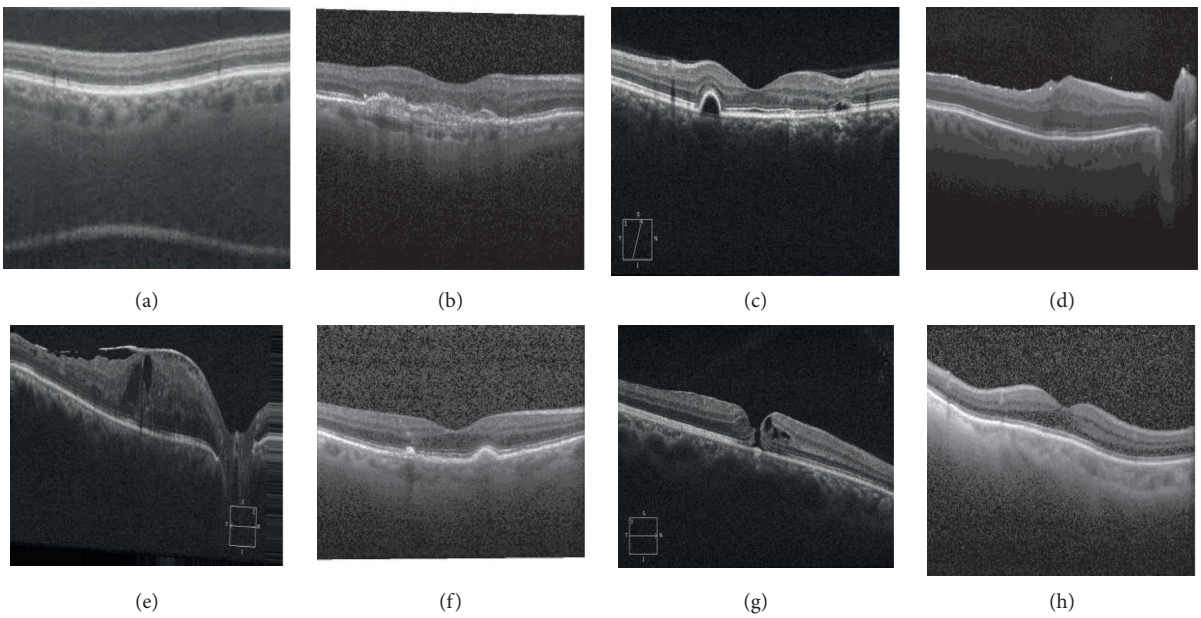


Figure 1: Sample OCT images. (a) AMD. (b) CNV. (c) CSR. (d) DME. (e) DR. (f) Drusen. (g) MH. (h) Normal.

exhaustive searching technique, has been used in the majority of the research studies to determine the ideal values)

(iv) Exploring the transfer learning of pretrained CNNs with an optimum set of hyperparameters

(v) Analyzing the performance of the variants of pretrained CNNs through rigorous simulations

(vi) A comparison of the performance of traditional and contemporary CNN architectures in terms of accuracy, precision, recall, and $F1$-score

We organize the rest of the article as follows: Section 2 discusses recent efforts relating to deep learning based retinal disease detection. In Section 3, we get into the intricacies of the dataset, deep neural network architectures, fine-tuning procedures, and Bayesian optimization. Experimental setup, adjusting the hyperparameters, and performance measures are discussed in Section 4. Following that, the experimental results and findings from the results are presented in Section 5. An in-depth analysis of error/misclassification of the images is also presented in Section 5. Finally, in Section 6, we summarize and conclude our study.

2. Literature Survey

The segmentation and thickness of retinal layers in OCT images are used to detect and diagnose retinal diseases. Changes in retinal layers owing to any disease are uncommon, and interpreting the data without a specialized benchmark technique is impossible. Standard image processing algorithms for detecting abnormalities in the retinal layer have certain drawbacks, such as being time-consuming and requiring sufficient subject knowledge. It is also challenging to generalize the procedure for automatic processing [29, 30]. With the evolution of technology and the introduction of Artificial Intelligence (AI), many researchers have begun to use deep learning based CNN to detect retinal diseases in OCT images. The application of CNN-based deep learning models is the topic of this review.

Obata [20] used deep learning and multivariate models to construct a model for predicting MH using preoperative OCT images and obtained precision of 46% and 40%, respectively. Hassan [18] used pretrained deep CNN to construct a system for reliable and automatic CSR detection from OCT images. For categorization, the authors used CNN models like AlexNet, ResNet-18, and GoogleNet. A statistical evaluation of parameters has been used to compare the performance of deep CNN, and AlexNet's classification accuracy from OCT Image Database was 99.64%. Subramaniam et al. [31] examined the most recent automated methods for detecting and classifying DR that used deep learning techniques. Binary classification, lesion-based classification, and vessel-based classification are some of the strategies used in this attempt. The publicly available fundus DR datasets have been provided, and deep-learning methodologies have been briefly explained. A work by [32] examined and analyzed the application of deep learning approaches at the various stages of DR detection using fundus images. This work included numerous parts of that pipeline, including widely used datasets, preprocessing approaches, and how they speed up and improve model performance, and the building of deep learning models for disease diagnosis and classification, as well as the localization of disease lesions.

The DeepDR system proposed by Dai [33] comprised three subnetworks: image quality evaluation, a subnetwork

with lesion-awareness, and a grading subnetwork for DR. This system was trained on fundus images and used a multitask network with transfer learning. Venkatasen et al. [22] identified that, in a pooling layer, the positional relations have been suppressed in classical CNN. Since the positional information from images can be learned by a capsule network, the authors sought to apply OCT images on a capsule network to overcome this issue and found that a capsule network can be replaced by a capsule network and enhanced classification accuracy. This method attained a classification accuracy of 99.6%, which is comparable to other methods published for CNV, DME, Drusen, and normal images. In an attempt by [24], three types of retinal diseases, namely, CNV, DMD, and DME, have been classified. The hyperparameters, such as the number of epochs, size of each batch, and optimizer type, have been modified using random search optimization for better performance in classifying various retinal diseases. The accuracy of this method was 97.01%. In an attempt by [23], it diagnosed CSC using a deep learning model, which was able to use OCT images to discriminate chronic from acute CSC. The authors found that the performance was comparable to that of ophthalmologists and was better than VGG-16 and ResNet50. This model had a 93.8% accuracy rate for CSC diagnosis.

To classify retinal OCT images, Li et al. [34] developed a classification technique based on an ensemble of four classification model instances based on ResNet50. On the retinal OCT dataset, this study applied a 10-fold cross-validation procedure. The proposed technique was found to have a classification accuracy of 97.3%, which is comparable to ophthalmologists with substantial clinical experience. Huang et al. [35] developed a layer-guided CNN that can distinguish between a healthy retina and prevalent macular diseases including Drusen, CNV, etc. Specifically, retinal layer segmentation maps have been created using an effective segmentation network that can distinguish between retinal layers linked with relevant retinal lesions. LGCNN then combines the data from two lesion-related layers using two well-designed subnetworks. The precision was believed to be around 88%. [13] suggested a deep learning based classifier for computer-assisted categorization of DME, Drusen, and CNV. This study used a six-layer deep CNN to perform the classification of the OCT images into four types, achieving an accuracy of 99.69%. [16] showed that utilizing a Generative Adversarial Network (GAN) to perform few-shot learning can increase the applicability of deep learning in OCT diagnosis of uncommon diseases. Four major classes with a large number of datasets and five classes of rare retinal diseases with a few-shot dataset have been considered for this study. The accuracy of this approach was 93.9%. [36] described a deep CNN architecture for effectively identifying and classifying patients into normal, DMD, and DME categories. The Kuan filter is used to remove speckles from raw OCT images to reduce intrinsic speckle noise. The classification accuracy of this work has been 95.7%. [11] used DenseNet-100 as a feature extractor, with CenterNet as a one-stage detector for localizing and classifying disease lesions. The authors tested the technique on two datasets, APTOS-2019 and IDRiD, and found that it had an average accuracy of more than 97%. While reviewing the present works, we find a work by Kaliappan et al. [19] that presented OCT images collected from the King Abdullah University Hospital in Irbid, Jordan. CNV, MH, CSR, Geographic atrophy, Macular Retinal Oedema, and Vitreomacular Traction are among the eye diseases by this collection, which includes 21,991 OCT images. A model based on the U-Net has been built to categorize where the images are of real Jordanian patients, and the annotation was done by ophthalmologists. This dataset was subjected to two classification tasks: a binary classifier that distinguishes between images from healthy eyes and diseased eyes (abnormal). The binary categorization was 84.9% accurate. Multiclass classification is the second classification challenge, in which the model is trained to discriminate between several diseases in addition to the normal condition, with 63.68% accuracy. In addition, a summary of existing deep learning models in glaucoma assessment utilizing OCT images has been provided by [14]. Apart from using deep learning models, image segmentation techniques and algorithms based on machine learning have also been proposed for image analysis [37–41].

As seen from the research attempts described above, deep learning architectures are increasingly being used in the diagnosis of retinal diseases from OCT images. However, various gaps in the usage of deep learning architectures that must be addressed include faster training times and fewer parameters. We employ transfer learning to reduce training time, and the optimal values for hyperparameters are chosen using Bayesian Optimization. Furthermore, we found that the current research works sought to detect 3 to 5 retinal diseases. We collected OCT images with AMD, CNV, DME, CSR, DR, Drusen, and MH diseases and made them publicly available to find a wide variety of retinal abnormalities.

3. Materials and Methods

The purpose of this study is to develop and compare a few models for diagnosing retinal diseases using OCT images using various CNN architectures. This section discusses all of the materials as well as the procedures used.

3.1. Datasets. The OCT images for retinal diseases acquired from Kaggle fall into the following categories: AMD, CNV, DME, DRUSEN, and NORMAL. We have also used images from OPEN-ICPSR, a no-cost, self-publishing resource for social, behavioral, and health sciences research data, to add a few more retinal diseases like CSR, DR, and MH. The OCT images collected from various sources such as Kaggle and Open-ICPSR are then augmented. Both datasets contain different classes and different numbers of images. To use this as input to a neural network model, the datasets have to be equalized. The greater the amount of data available to the network, the more features it will be able to learn. Image augmentation is a technique for artificially producing new training images. Rotation, flipping, cropping, and translation are examples of image augmentation techniques that have been used to assist lessen a model's overfitting. The dataset comprises around 24,000 images after augmentation, approximately 3000 images in each category. After

equalizing the data from both datasets, a new dataset called "Retinal OCT–C8" is developed and hosted as a public dataset on Kaggle. We have also used on-the-fly data augmentation to deliver real-time augmentation. That is, while a model is still being trained, it generates augmented images on the fly and ensures that each epoch of the model receives new variants of the images. The images that have been altered are not included in the original image corpus. If this was the case, the model would be continually exposed to the original images, leading to overfitting. Normalization of the image's size and format is a critical operation. All images have been resized to 224 ∗ 224 (for VGG16 and DenseNet201) and 299 ∗ 299 (for InceptionV3 and Xception) pixels at a resolution of 96 ∗ 96 dots per inch.

3.2. CNN. CNN architecture is a popular deep learning method for image classification and a key technique for modeling complex processing in applications with a lot of data. It is cutting-edge in image classification tasks and is programmed to extract visual patterns from input images directly. CNN is based on the work of Kunihiko Fukushima, a Japanese scientist who invented the Neocognitron, a very primitive image recognition neural network. The challenge of handwritten digit categorization has been effectively implemented by CNN with a gradient-based algorithm. It then became the state-of-the-art in a variety of object recognition tasks, and it is currently utilized in a variety of other fields, including object tracking and identification and text and action recognition. A significant property of CNN is its capacity to automatically learn hierarchical feature representations. Edge-based features are often detected by the first few layers of CNN. The early layers' output is sent into intermediate layers, which extract more complicated features like corners and edges. The layers recognize higher-level features like objects, faces, and so on as we progress deeper into the network. This means that the earliest layers' features are generic and can be used to solve a range of issues, but the latter layers' characteristics are particular to the dataset and task at hand. When compared to traditional feed-forward neural networks, CNN has the advantage of requiring fewer neurons and hyperparameters. For image recognition applications, several baseline CNN architectures have been created and effectively utilized to complex visual imagery problems. To develop the proposed models in this work, we used pretrained models including VGG16, DenseNet201, InceptionV3, and Xception. In the next section, we will go over these ground-breaking CNN designs.

3.3. VGG16. VGG16 is a 16-layer network presented in 2014 by Simonyan and Ziserman of Oxford University's Visual Geometry Group Lab [28], It is much deeper than AlexNet but has a simpler network, because huge kernel-size filters are replaced with multiple 3 ∗ 3 kernel-size filters. VGG16 is made up of thirteen convolutional layers and three fully connected layers. Figure 2 depicts the architecture of VGG16. VGG19, an improved version of VGG16, has sixteen convolutional layers and three fully connected layers.

3.4. DenseNet201. DenseNet201 [42] is a CNN that employs dense blocks to establish dense connections between layers, with all levels being linked directly. Each layer in a feed-forward technique is linked to every other layer. When a layer is generated, the feature maps of all previous layers are regarded as independent inputs for each layer, whereas the feature maps of the current layer are passed on as inputs to all subsequent layers. The elimination of the vanishing-gradient problem, improved feature propagation, feature reuse, and a large reduction in the number of parameters are all advantages of DenseNets. For these reasons, we chose to create a model using this CNN variation. Figure 3 depicts a dense 5-layer block.

3.5. Inception. InceptionV1 is a deep convolutional architecture that was launched as GoogLeNet by [43]. The Inception design was later modified in several ways, the first of which was the addition of batch normalization [44]. This is named InceptionV2. InceptionV3 [45] includes label smoothing, factorized 7×7 convolutions, and a classifier for transferring label information deeper down the network. The model's symmetrical and asymmetrical building components include convolutions, max pooling, concerts, dropouts, and fully connected layers. The softmax function is part of the InceptionV3 architecture's last layer, which includes 48 layers in total and an input layer that accepts images with a resolution of 299×299 pixels. While preserving speed and accuracy, InceptionV3 considerably cuts processing expenses.

3.6. Xception. Xception is a variation of the Inception architecture that uses depth-wise separable convolutions instead of the standard Inception modules. The Xception architecture [46] has 36 convolutional layers as its feature extraction base and except for the first and last layers, the convolution layers are divided into 14 modules, each of which is surrounded by linear residual connections. In a nutshell, the Xception architecture is a residually connected depth-wise separable convolution layer stack. This model substituted depth-wise separable convolutions for standard inception modules, which were preceded by a point-wise convolution (1 ∗ 1). In most traditional classification problems, the Xception architecture outperformed VGGNet, ResNet, and InceptionV3.

3.7. Transfer Learning. Transfer learning has recently piqued the interest of researchers. It is an approach for fine-tuning previously trained neural networks to create new AI models [47]. In other words, it uses established knowledge to address distinct but similar domain issues. Its goal is to complete information transmission between related areas, and it has become extremely popular because it cuts training time and utilizes far fewer data to increase performance. Transfer learning is typically portrayed in computer vision through the use of pretrained models. A pretrained model has been trained on a large benchmark dataset to handle a problem similar to the one at hand. Importing and using

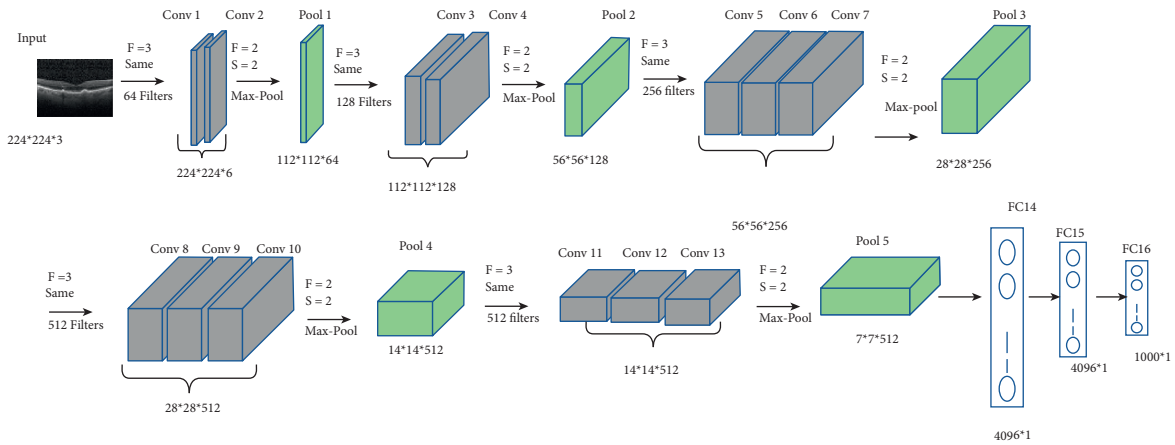


Figure 2: VGG16 architecture.

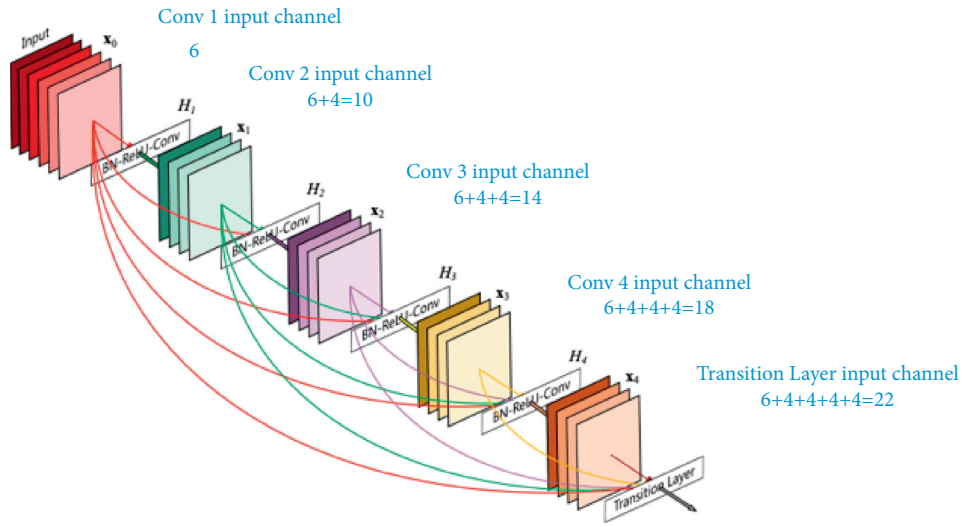


Figure 3: A 5-layer dense block [43].

models that have previously been tested and published is one technique to reduce the computational expense of training new models (e.g., VGGNet, Inception, and Exception). Canziani et al. [48] used the ImageNet dataset to investigate the performance of pretrained models on computer vision challenges. Using transfer learning, large CNNs are used to train several pretrained models. We repurpose the pretrained CNN versions VGG16, DenseNet201, InceptionV3, and Xception for our dataset by removing the classifier and adding a few classification layers, as well as retraining the top layers of convolution base. Here is what they are:

Training the classifier (feature extractor) by freezing the convolutional base: we can preserve the convolution base in its original form while using ImageNet weights. The classifier produces 1000 different output labels in pretrained models; however, the number of neurons in the output layer can be determined by the number of classes in our dataset. As a result, we may import the convolutional base and add our classifier to it. The classifier receives the output from the convolutional base. The pretrained model can be used as a feature extractor in this approach.

Fine-tuning a few top layers: we maintain the weights of the initial layers frozen and retrain the higher layers to learn the dataset-specific features since the lower layers correspond to general features (dataset independent features) and the higher layers refer to unique features (dataset dependent features). Pretrained models are used as a fine-tuner in this case.

Since the pretrained models trained on ImageNet have been used to identify retinal syndromes in many of the attempts described in Section 2, we also employ transfer learning to fine-tune our models using the above two ways. Even though we have roughly 24000 OCT images for training and testing the models, this may not be enough for deep neural networks, resulting in overfitting. When the target data set is tiny, the main benefit of transfer learning becomes apparent. The model may be prone to overfitting in many circumstances, and augmentation may not necessarily solve the overfitting problem. As a result, transfer learning

has been deployed. We can also save a substantial amount of training time by relying on existing knowledge because the weights learned from ImageNet datasets have been used. Retraining a model from scratch demands the random selection of all weights, which takes huge computational power and time. Hence, training the classifier and the convolution base's top layers should be sufficient. Here is a rundown of the transfer learning techniques employed in this study:

(i) Pick one model at a time from the list of pretrained models

(ii) Add classification layers according to the dataset and pretrained models

(iii) Train the model using strategies 1 and 2 in turn

(iv) Out of all the pretrained models, find the strategy with the highest accuracy

The details and results of implementing these strategies are discussed in Sections 4 and 5.

3.8. Hyperparameters Optimization. Hyperparameters are network parameters that define the structure of the network, such as the hidden units' size, dropout, activation function, and weight initialization, as well as how the network parameters such as learning rate, momentum, batch size, and epochs are trained. The process of identifying the best settings for hyperparameters in a learning algorithm is known as hyperparameter tuning. Hyperparameter tuning is to identify optimal values for hyperparameters to reduce the loss function and improve results. The various optimization techniques include manual search, grid search, random search, and Bayesian Optimization. A random search produces hyperparameter combinations at random, tries to fit the dataset, and assesses its accuracy. It is possible that certain configurations that would have been ideal were overlooked. While random search is quicker, it may not always produce the best results. In manual search, using previous experience, we select hyperparameters for a model. The model is then trained and evaluated using these parameters. This approach is repeated with a different set of values for the same hyperparameters until maximum accuracy is acquired or the model has attained the optimal error. Because manual search is biased and comprehensive, it may not be the best option. In grid search, the same procedure as random search is used for tuning the hyperparameters, but with one exception. Each hyperparameter combination is tried. This adds time to the process and makes it inefficient. However, it is the most successful since the best option is less likely to be overlooked.

Unlike grid and random searches, Bayesian optimization takes advantage of previous iterations of the algorithm. Each hyperparameter guess is independent in the grid and random searches. With Bayesian techniques, on the other hand, we move closer to the perfect solution with each selection and testing of alternative hyperparameters [49, 50]. When it comes to identifying the optimum potential hyperparameter settings, Bayesian optimization algorithms surpass grid and random searches. Because of the amount of data and computing density, more time is needed to train deep learning models. Bayesian optimization can be quite useful in these situations. In this work, we employ Bayesian optimization to optimize the hyperparameters of the classifier layers in conjunction with pretrained models. To summarize, we suggest using CNNs powered by transfer learning and Bayesian optimization to create a few models to classify OCT images. The workflow of the proposed models is depicted in Figure 4. A few classifiers have been developed using the pretrained models and retrained using transfer learning for feature extraction. While fine-tuning, optimum number of convolution base layers to be retrained has been found. The ideal values for hyperparameters have been determined via Bayesian optimization. For the set of values of hyperparameters, the pretrained models have been trained using the training dataset and stored as checkpoints. The models with the ideal hyperparameter values that provided the highest accuracy have been evaluated using the validation dataset. Finally, using the testing dataset, the performance of classifiers is assessed.

4. Details of Experiments

We conducted two sets of experiments, namely, Feature Extractor and Fine-tuner, to examine the performance of the models developed, and the details of experiments are given below.

4.1. Experimental Platform. We imported the necessary Keras model architectures and instantiated them with ImageNet weights. Since the developed models consume a lot of power and require high-performance hardware to function properly, we ran the proposed models on Graphical Processing Units (GPUs). The hardware and software configurations utilized are listed in Table 1.

4.2. Tuning of Hyperparameters. In deep learning algorithms, hyperparameters are significant because they specify training details and have a direct impact on model output [51]. Choosing the appropriate hyperparameter settings is critical. We have used Bayesian Optimization to obtain the best values for hyperparameters while maintaining excellent accuracy in this study. It is a method for determining the lowest or maximum of an objective function. In this study, we wish to maximize the accuracy and use the Gaussian Process (GP) as the probabilistic model [50]. GP generates a hypothesis for unknown parameters based on previously known parameters. Although the Bayesian approach takes longer to select hyperparameters, it takes less time to assess the objective function, resulting in low computational costs. Table 2 summarizes the hyperparameters tuned in our work, as well as their search space.

4.3. Experimental Setup. After the dataset has been randomized, the training and testing datasets have been split, with 70% of the dataset being used to train the classifier, 15% being used for validation, and 15% being used for testing. This is done to guarantee that as much data as possible is

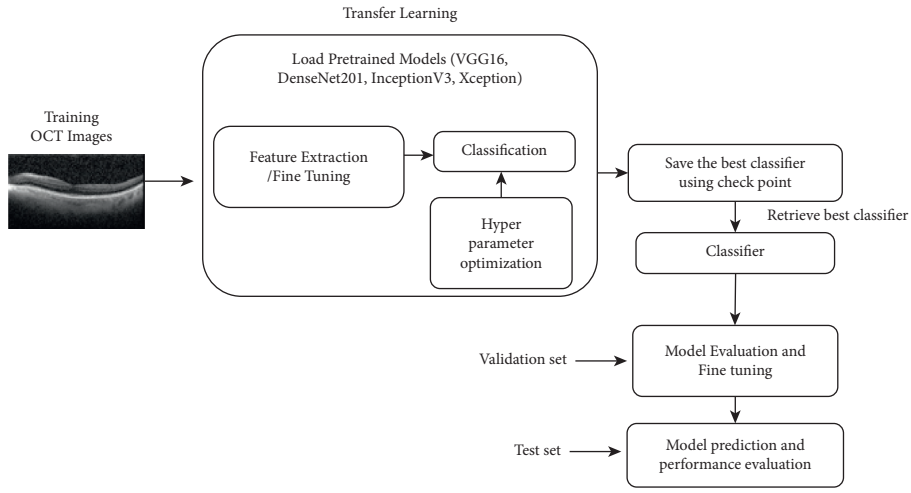


Figure 4: Workflow for the proposed classifiers.

Table 1: Experimental platform.

Item name	Specifications
GPU	GPU DELL EMC 740
RAM	128 GB
GPU RAM	32 GB
DISK	4 TB
OS	Ubuntu
Language	Python
IDE	Jupyter notebook environment

available for training, resulting in a more accurate model. The training and validation datasets have been used to train and fit the model, while the test set has been used to evaluate the model's prediction performance on samples it had never seen before. For both sets of experiments, we downsized all images to $224 * 224 * 1$ and $299 * 299 * 1$ and used in-place image augmentation to accommodate the input of the developed models. In all of the models, the categorical cross-entropy is employed as the loss function. We ran the models for 75 epochs but stopped them early. Early stopping is a technique, in which the model is trained for an arbitrary number of epochs and then stopped when the validation accuracy or validation loss does not improve. To monitor the validation accuracy, we employed early stopping and set patience to 5, which helps quit the training if the validation accuracy does not improve. Another reason for early stopping is that it allows us to terminate the training process when the model becomes overfit. To conduct the proposed tests, we removed the classifier layer from these models and replaced it with our own. For each of the pretrained models, Table 3 shows the number of fully connected layers added to the classification block.

The actual output layer in all of the pretrained models is a 1000-class softmax activation. This layer is replaced with an eight-category softmax layer. The number of neurons in each fully connected layer is a configurable hyperparameter. Pretrained models such as VGGNet, DenseNet201, InceptionV3, and Xception have been utilized as feature extractors in the first set of tests, and the retrieved features have been then used to train the newly added classifier. The weights learned from the ImageNet dataset have been used in the convolution base. In the second set of tests, we retrained a few top layers of the convolution base. Bayesian optimization has been used to find the best values for the hyperparameters. A total of 20 iterations of Bayesian optimization have been performed. We set the number of epochs to 75 for each iteration of Bayesian optimization. Each iteration's accuracy and loss have been recorded. Table 4 shows the hyperparameter settings that resulted in the highest accuracy for all models. The hyperparameters found within 20 iterations were deemed best in our study because more iterations did not result in substantial changes. If we employ a vast search space, we can get a better set of values for hyperparameters, but at the expense of a huge computation time.

4.4. Performance Metrics. Following the development of the models, the next step is to evaluate their effectiveness using metrics against the test datasets. The developed CNN models have been evaluated using a variety of performance measures, including accuracy, precision, recall, *F*1-Score, macro, and weighted average. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) indices have been used to calculate the values for the performance metrics. TP stands for the total number of correctly classified images in each class. FP stands for the number of images misclassified in all other classes except the correct one. FN stands for the number of images misclassified in the relevant class. The number of images correctly identified in all other classes except the correct one is referred to as TN. Equations (1) to (4) are used to determine TP, FP, TP, and TN, with $i = 1,2,3$, and 4 signifying the four classes.

$$tp_i = c_{ii}, \tag{1}$$

$$fp_i = \sum_{l=1}^{n} (c_{li}) - tp_i, \tag{2}$$

Table 2: Hyperparameters and their search space.

Parameter	Search space	Description
Optimizer	Adam, RMSProp, SGD, AdaDelta	To optimize the input weights by comparing the prediction and the loss function
Learning rate	$1e-3$, $1e-4$, $1e-5$, $1e-6$	To determine the step size at each iteration while minimizing the loss function
Activation function	ReLu, Elu and Tanh, Leaky ReLu	To introduce nonlinearity into the output of neurons
Number of neurons in customized layers	64,128, 56, 512,1024	To compute the weighted average of the input
Batch size	32,64,128	Number of training examples utilized in one iteration

Table 3: Modification of the classification block.

Pretrained models	Number of layers added
VGG16	2 fully connected + 1 softmax
DenseNet201	2 fully connected + 1 softmax
InceptionV3	1 fully connected + 1 softmax
Xception	1 softmax

$$fn_i = \sum_{l=1}^{n} (c_{il}) - tp_i, \tag{3}$$

$$tn_i = \sum_{l=1}^{n} \sum_{k=1}^{n} (c_{lk}) - tp_i - fp_i - fn_i. \tag{4}$$

Accuracy is defined as the number of samples properly identified as belonging to a specific class divided by the total number of samples in that class and is calculated by

$$\text{accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}. \tag{5}$$

The number of samples correctly categorized as a certain class out of the total number of actual samples in that class is defined as recall and is computed using

$$\text{recall} = \frac{(\text{TP})}{(\text{TP} + \text{FN})}. \tag{6}$$

Precision is defined as the number of samples accurately categorized as a specific class out of the total number of samples categorized as that class and is given by

$$\text{precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}. \tag{7}$$

*F*1-Score is defined as the harmonic average of the precision, and recall, that is, the weighted average of Precision and Recall. It is calculated as in

$$\text{F1} - \text{score} = \frac{(2 * \text{precision} * \text{recall})}{(\text{precision} + \text{recall})}. \tag{8}$$

The unweighted average of the class-wise scores is used to determine the macro average. To get the final averaged metric, macro-average gives equal weight to each of the eight classes in the dataset. The weighted average is computed by taking the weighted average of class-wise scores, with the weights proportional to the number of instances of each class; that is, the contribution of each class to the average is weighted by its size.

5. Experimental Results and Findings

We examined the performance of the pretrained CNN models used in this study, including VGG16, DenseNet201, InceptionV3, and Xception, as feature extractors and fine-tuners. The experiments have been conducted with the tuned hyperparameters listed in Table 4, which generated the good results during training. The classification report for the proposed models is presented in Tables 5–8. Due to the uncertain nature of images, the models may result in better accuracy but fail to realize the images properly and hence may perform poorly when the images are varied. This indicates that the models are not robust enough and hence limit their usage. So, accuracy is not enough alone for classification task. We need to look at some other metrics to make sure our models are reliable. Because the number of images in each class in the test data set is roughly similar, the weighted and macro averages for each model are nearly identical.

Table 9 provides a comparison of the performance of the models developed in this work.

Since we need to classify retinal diseases into one of eight categories, we assess the performance of all the models in the dataset against each of the eight classes. For this, we utilized equations (1) to (4) to calculate the indices TP, FP, TN, and FN. We use the confusion matrix obtained during testing to calculate the values of these indices. As we all know, the confusion matrix is a visualization tool to know how well-predicted classes match the actual classes. The confusion matrix acquired when testing VGG16 is shown in Figure 5. The diagonal elements represent correct classifications. On the other hand, the rest are misclassifications. The *X*-axis depicts predicted classes, whereas the *Y*-axis depicts actual classes. For example, VGG16 predicted 20 images with DME disease as CNV, 1 DME image as AMD, and so on, as shown in Figure 5(a).

5.1. Performance Comparison with Other Models. The performance of the models developed in this work is further evaluated by comparing them to other similar models that categorize retinal diseases. The comparison results are summarized in Table 10.

Table 4: Hyperparameters with tuned values.

Hyperparameters	VGG16		DenseNet201		InceptionV3		Xception	
	Feature extractor	Fine-tuner	Feature extractor	Fine-tuner	Feature extractor	Fine-tuner	Feature extractor	Fine-tuner
Optimizer	Adam	Adam	Adam	Adam	RMSProp	RMSProp	Adam	Adam
Learning rate	0.0001	0.0001	0.00001	0.00001	0.00001	0.0001	0.001	0.0001
Activation	tanh	tanh	elu	relu	tanh	relu	elu	relu
No. of neurons	256	512	128	128	64	512	128	256
Batch size	32	32	32	32	32	32	32	32

Table 5: Performance of VGG16.

Class labels	Precision (%)		Recall (%)		F1-score (%)	
	Feature extractor	Fine tuner	Feature extractor	Fine tuner	Feature extractor	Fine tuner
AMD	99.41	100	96	99.71	97.67	99.86
CNV	79.74	87.24	69.71	95.71	74.39	91.28
CSR	83.05	97.77	96.57	100	89.3	98.87
DME	73.83	93.49	62.86	90.29	67.90	91.86
DR	81.44	99.14	84	99.14	82.7	99.14
Drusen	67.19	95.72	60.86	83.14	63.89	88.99
MH	95.41	100	77.14	98.29	85.31	99.14
Normal	62.65	89.81	87.71	95.71	73.1	96.67
Macro average Weighted average	80.34	95.4	79.36	95.25	79.28	95.23

Table 6: Performance of DenseNet201.

Class labels	Precision (%)		Recall (%)		*F*1-score (%)	
	Feature extractor	Fine tuner	Feature extractor	Fine tuner	Feature extractor	Fine tuner
AMD	97.34	100.00	94.82	100.00	95.12	99.12
CNV	91.08	99.82	92.45	98.12	92.91	89.66
DME	94.26	99.55	89.12	99.34	90.72	99.61
CSR	97.18	99.11	93.29	97.56	94.61	98.55
DR	89.99	99.79	96.51	98.69	94.95	97.99
Drusen	91.73	99.51	93.21	98.88	92.91	97.99
MH	91.00	100.00	96.57	99.11	93.98	98.44
Normal	92.09	99.91	94.12	98.99	94.01	98.03
Macro average	93.08	99.71	93.76	98.84	93.65	97.42
Weighted average	93.08	99.71	93.76	98.84	93.65	97.42

Table 7: Performance of InceptionV3.

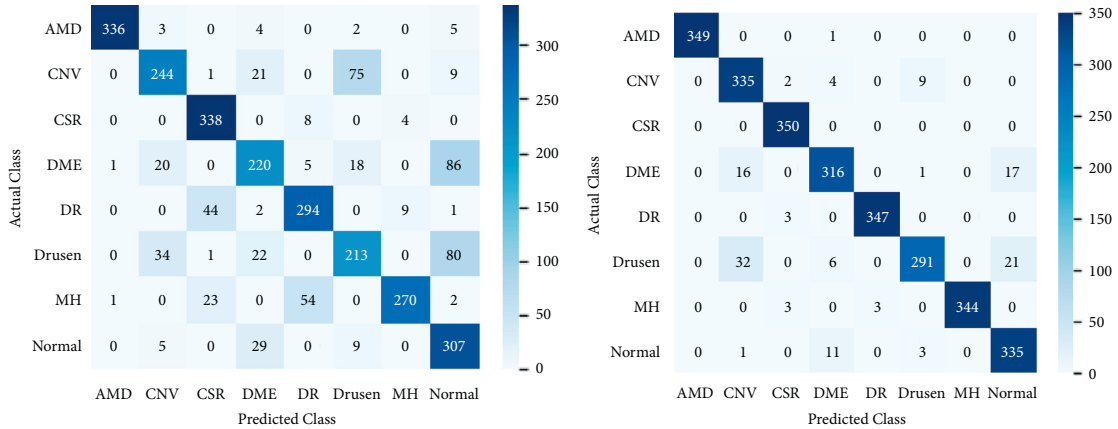
Class labels	Precision (%)		Recall (%)		*F*1-score (%)	
	Feature extractor	Fine tuner	Feature extractor	Fine tuner	Feature extractor	Fine tuner
AMD	90.12	95.62	91.81	92.31	90.60	93.27
CNV	88.15	96.71	90.81	95.91	87.96	90.06
DME	90.31	93.98	87.48	93.27	87.72	91.72
CSR	88.18	95.91	91.02	91.78	90.34	89.45
DR	87.99	93.11	90.18	93.91	90.06	91.62
Drusen	89.41	96.81	89.38	89.95	89.43	93.02
MH	90.01	96.21	91.64	96.81	89.23	91.62
Normal	88.62	95.47	89.74	95.99	90.82	93.49
Macro average Weighted average	89.10	95.48	90.26	93.74	89.52	91.78

Table 8: Performance of Xception.

Class labels	Precision (%)		Recall (%)		*F*1-score (%)	
	Feature extractor	Fine tuner	Feature extractor	Fine tuner	Feature extractor	Fine tuner
AMD	92.17	98.23	89.23	98.56	90.12	96.99
CNV	90.56	98.18	91.10	97.34	90.06	95.61
DME	89.91	96.99	88.97	98.13	89.97	97.03
CSR	88.65	97.26	90.45	96.99	91.22	98.41
DR	92.11	98.61	89.34	95.09	90.23	96.21
Drusen	91.99	97.49	92.31	95.01	91.24	95.24
MH	92.05	98.12	91.81	96.38	90.03	95.02
Normal	90.51	94.78	89.45	95.73	91.45	96.12
Macro average Weighted average	90.99	97.46	90.33	96.65	90.54	96.33

Table 9: Validation and testing accuracy of the proposed models.

Experiment scenario	VGG16		DenseNet201		InceptionV3		Xception	
	Valid (%)	Test (%)	Valid (%)	Test (%)	Valid (%)	Test (%)	Valid (%)	Test (%)
Feature extractor	80.64	79.36	94.57	93.81	91.63	89.73	92.11	90.99
Fine-tuner	95.21	95.25	99.23	99.71	96.92	96.78	98.12	97.92



(a) (b)

Figure 5: Confusion matrix. (a) VGG16 (feature extractor). (b) VGG16 (fine tuner).

From Table 10, we can see that the test accuracy of the proposed models is relatively good compared with the accuracy of other recent deep learning methods. One good thing about the proposed models is that they all attempt to identify more retinal diseases from OCT images than with other approaches.

6. Results and Discussion

Table 9 presents the overall validation and testing accuracy for each of the models across all eight classes. As shown in Table 9, when employing the pretrained models as feature extractors and training the classifier using the extracted features, the validation and testing accuracy is lower than that when retraining a few top layers. The fine-tuning technique keeps the pretrained models' weights on earlier levels and retrains them on the top layers. But, in strategy 1, the pretrained models have been just employed as feature extractors, with no fine-tuning, and the top layers are not retrained exclusively for our dataset. As a result, the features learned are unique to ImageNet, and so the accuracy is lower than strategy 2. But, interestingly, the three models DenseNet201, InceptionV3, and Xception achieved high accuracy for strategies 1 and 2, showing that optimization allowed for improved generalization in these models. VGG16, on the other hand, comparatively has low accuracy. We may be able to improve accuracy by retraining this model for more epochs. We stopped training the models since we employed early stopping, and there was no change in validation accuracy after 5 iterations. We experimented by removing early stopping and found that there is an increase in accuracy in VGG16 as well. While analyzing the reason

Table 10: Comparison of proposed models with other deep learning models.

Models	Retinal diseases	Classification accuracy (%)
Models proposed in the literature		
OctNET [13]	DME, CNV, and Drusen	99.7
Layer guided CNN [35]	DME, CNV, and Drusen	89.9
GAN [16]	DME, CNV, MH and Drusen	93.9
Deep CNN [36]	DMD and DME	95.7
CenterNet [11]	DR	98.1
AlexNet, ResNet-18, GoogleNet [18]	CSR	99.6
Capsule network [22]	DME, Drusen, and CNV	99.6
CNN [24]	DMD, DME, and CNV	97.0
Deep CNN [23]	CSR	93.8
Proposed pretrained models in this work		
VGG16		
(a) As a feature extractor		79.36
(b) As a fine tuner		95.25
Densenet201		
(a) As a feature extractor		**93.81**
(b) As a fine tuner	AMD, CNV, DME, CSE, DR, Drusen, MH	**99.71**
InceptionV3		
(a) As a feature extractor		89.73
(b) As a fine tuner		96.78
Xception		
(a) As a feature extractor		90.99
(b) As a fine tuner		97.92

for the high accuracy of DenseNet201, we found that better feature reusing capability leads to high accuracy. In addition, DenseNet201 alleviates the vanishing-gradient problem, supports feature propagation, and substantially reduces the number of parameters. Nevertheless, this model requires a large amount of GPU memory for convolution operation. Based on our review of the literature, we found that a few attempts used GPUs to train the models. But we are unable to compare the performance of the proposed models with these models in terms of training time due to the differences in GPU configuration and disparity in the datasets. Because we retrain a few top layers of the models in addition to the classifier component, training as feature extractors takes less time than training as a fine-tuner. This clearly indicates that retraining the entire model would take much more time. Hence, it is evident that transfer learning reduces the training time. Among the developed models, even though DenseNet201 gives the highest accuracy, it takes huge time to train the models, as it has a huge number of layers. We further calculated the number of parameters retrained in each of the developed models, and the results are presented in Table 11.

Among all the models, DenseNet201 retrained a smaller number of parameters. Nevertheless, the time taken by this model for training is large as it has a huge number of layers. We enumerated a set of research issues to be addressed by the proposed effort in Section 1. Now, we will take a look at how the proposed models have addressed these issues. In this attempt, four pretrained models trained on the ImageNet dataset have been used to develop models for detecting retinal diseases from OCT images. These pretrained models are simple to use and produce better results with less training effort because they provide the architecture for free. During transfer learning in the proposed study, the pretrained models have been deployed with a few alterations on a new classification task. This resulted in higher accuracy than constructing models from the ground up. Table 10 shows how this works. Using fine-tuning, we can use pretrained networks to recognize classes in new datasets that they were not trained on before. Fine-tuning was more accurate than transfer learning via feature extraction because the weights of the later layers were retrained on the dataset used in the study. Bayesian optimization was then used to find the ideal values for the hyperparameters, which resulted in a considerable improvement in performance overutilizing the default values for the hyperparameters. To summarize, we believe that the following factors, when compared to other models, contribute to improved accuracy.

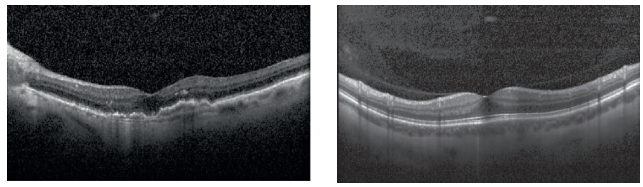
(i) Finding optimal values for the hyperparameters via Bayesian optimization

(ii) Using transfer learning to fine-tune the top layers of the convolutional base

6.1. Error Analysis. To understand the challenges of this task, we carried out further analysis of the errors made by our models. Error Analysis refers to the process of examining test set images that the models misclassified so that we can understand the underlying causes of the errors. A classification model's results on new images can be categorized into one of four categories, namely, true positives, false positives, true negatives, and false negatives. True or false refers to whether the predicted class matches the actual class in all four cases, and positive or negative refers to the classification the model has assigned to observation. For instance, in the confusion matrix for the VGG16 model, we can find that the

Table 11: Trainable parameters in proposed models.

Model	Number of parameters retrained	
	Feature extractor (M)	Fine-tuner (M)
VGG16	4.7	5.5
Xception	8.3	14.4
InceptionV3	1.1	2.3
DenseNet201	2.31	3.9



(a) (b)

Figure 6: Error analysis. (a) An OCT image with CNV disease. (b) An OCT normal image.

true positive for the AMD class is 336. This indicates that out of 350 AMD samples, 336 samples have been classified as AMD, and 14 instances have been misclassified not AMD. Similarly, for CNV, only 244 instances have been classified correctly as CNV, and 106 instances have been misclassified as not CNV. Below we discuss a few cases. Figure 6 shows an OCT image of CNV retinal disease and a normal image.

The actual class for the image in Figure 6(a) is CNV. But the VGG16 model has predicted this image as normal, that is, without any retinal disease. Although the feature map strongly highlights the presence of symptoms for retinal disease, we cannot immediately be sure that this is the reason for misclassification. But the image is correctly classified by other models. Similarly, an OCT normal image has been predicted by VGG16 as having CNV disease. One reason may be that the VGG16 model has not fetched the features from the images properly. So, the details that led to the misclassification must be found. As a result, we believe that the misclassification data can be used to increase classification accuracy. Assume that images are frequently misclassified with many classes for one single class. Instead of considering all image classes, we should focus on specific misclassified classes to mine important information.

7. Conclusion and Future Work

Retinal diseases have become a major public health concern in recent years and accurate detection is a challenge. Manual localization of retinal disease requires the use of trained human experts to detect finer points of interest in OCT images and classify them into the relevant disease using a grading system. Automated retinal disease detection models are necessary to overcome the obstacles of manual detection, and this work investigated the application of deep learning models to diagnose retinal diseases using OCT images. Transfer learning has been chosen for this research because it has the following advantages: (i) no need for excessively large training datasets; and (ii) only the weights of a few top layers need to be retrained, requiring little processing effort. Since developing a model from scratch requires a lot of computational power, we used pretrained models, such as VGG16, DenseNet201, InceptionV3, and Xception as feature extractors and fine-tuners.

With exception of VGG16, all other models showed comparable accuracy to other deep learning models, when using them as classifiers. When fine-tuned, however, they achieved an accuracy of over 95%. Because DenseNet201 is the deepest of all the pretrained models used in this study, it takes longer training epochs to achieve high accuracy. Additionally, Bayesian optimization was used to select the best values for hyperparameters used during training. The findings of this study led us to believe that using pretrained models based on Bayesian Hyperparameter optimization and transfer learning for the classification of retinal diseases from OCT images is a promising alternative. As a result, this research can be extended to detect a variety of additional retinal diseases and construct a few more deep learning models with fewer parameters and less training time. There is a trade-off between the selection of hyperparameters and the training time. Hence, we plan to further explore hyperparameters used in the optimization process. In the meantime, the trained model could be used with mobile devices to assist health practitioners to make fast and precise decisions about retinal diseases.

Data Availability

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

Conflicts of Interest

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

References

[1] N. Eddy, "FDA clears GE Healthcare's AI platform for X-ray scans," 2019, https://www.healthcareitnews.com/news/.

[2] V. E. Sathishkumar and Y. Cho, "Cardiovascular disease analysis and risk assessment using correlation based intelligent system," *Basic and Clinical Pharmacology and Toxicology*, vol. 125, no. 1, p. 61, 2019.

[3] L. Alzubaidi, "MedNet: pre-trained convolutional neural network model for the medical imaging tasks," 2021, https://arxiv.org/abs/2110.06512.

[4] M. S. Islam, J.-K. Wang, S. S. Johnson, M. J. Thurtell, R. H. Kardon, and M. K. Garvin, "A deep-learning approach for automated OCT en-face retinal vessel segmentation in cases of optic disc swelling using multiple en-face images as input," *Translational Vision Science & Technology*, vol. 9, no. 2, p. 17, 2020.

[5] S. Malliga, "Automated skin disease identification using deep learning algorithm," *International Journal of Advanced Science and Technology*, vol. 29, no. 3, pp. 255–260, 2019.

[6] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can machine-learning improve cardiovascular risk prediction using routine clinical data?" *PloS one*, vol. 12, no. 4, Article ID e0174944, 2017.

[7] B. van Ginneken, "Fifty years of computer analysis in chest imaging: rule-based, machine learning, deep learning," *Radiological physics and technology*, vol. 10, no. 1, pp. 23–32, 2017.

[8] V. Gulshan, L. Peng, M. Coram et al., "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *JAMA*, vol. 316, no. 22, pp. 2402–2410, 2016.

[9] G. Quellec, K. Charrière, Y. Boudi, B. Cochener, and M. Lamard, "Deep image mining for diabetic retinopathy screening," *Medical Image Analysis*, vol. 39, pp. 178–193, 2017.

[10] E. Rahimy, "Deep learning applications in ophthalmology," *Current Opinion in Ophthalmology*, vol. 29, no. 3, pp. 254–260, 2018.

[11] T. Nazir, M. Nawaz, J. Rashid et al., "Detection of diabetic eye disease from retinal images using a deep learning based CenterNet model," *Sensors*, vol. 21, no. 16, p. 5283, 2021.

[12] D. S. Kermany, M. Goldbaum, W. Cai et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[13] A. Sunija, "Octnet: a lightweight cnn for retinal disease classification from optical coherence tomography images," *Computer Methods and Programs in Biomedicine*, vol. 200, Article ID 105877, 2021.

[14] A. R. Ran, C. C. Tham, P. P. Chan et al., "Deep learning in glaucoma with optical coherence tomography: a review," *Eye*, vol. 35, no. 1, pp. 188–201, 2021.

[15] A. Patel, O. V. Swathika, U. Subramaniam et al., "A practical approach for predicting power in a small-scale off-grid photovoltaic system using machine learning algorithms," *International Journal of Photoenergy*, vol. 2022, Article ID 9194537, 21 pages, 2022.

[16] T. K. Yoo, J. Y. Choi, and H. K. Kim, "Feasibility study to improve deep learning in OCT diagnosis of rare retinal diseases with few-shot classification," *Medical, & Biological Engineering & Computing*, vol. 59, no. 2, pp. 401–415, 2021.

[17] N. Rajagopalan, V. Narasimhan, S. Kunnavakkam Vinjimoor, and J. Aiyer, "Deep CNN framework for retinal disease diagnosis using optical coherence tomography images," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 7, pp. 7569–7580, 2021.

[18] S. A. E. Hassan, "Deep learning-based automatic detection of central serous retinopathy using optical coherence tomographic images," in *Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*, April, 2021.

[19] S. Kaliappan, R. Saravanakumar, A. Karthick et al., "Hourly and day ahead power prediction of building integrated semitransparent photovoltaic system," *International Journal of Photoenergy*, vol. 2021, Article ID 7894849, 8 pages, 2021.

[20] S. Obata, "Prediction of postoperative visual acuity after vitrectomy for macular hole using deep learning–based artificial intelligence," *Graefe's Archive for Clinical and Experimental Ophthalmology*, vol. 260, pp. 1–11, 2021.

[21] S. V. Kogilavani, J. Prabhu, R. Sandhiya et al., "COVID-19 detection based on lung ct scan using deep learning techniques," *Computational and Mathematical Methods in Medicine*, vol. 2022, Article ID 7672196, 13 pages, 2022.

[22] M. Venkatasen, S. K. Mathivanan, P. Mani et al., "Effectiveness of contact tracing using KNN for COVID-19," *Journal of Mobile Multimedia*, vol. 17, pp. 789–808, 2021.

[23] V. E. Sathishkumar, J. Park, and Y. Cho, "Seoul Bike Trip duration prediction using data mining techniques," *IET Intelligent Transport Systems*, vol. 14, no. 11, pp. 1465–1474, 2020.

[24] V. E. Sathishkumar, J. Park, and Y. Cho, "Using data mining techniques for bike sharing demand prediction in Metropolitan city," *Computer Communications*, vol. 153, pp. 353–366, 2020.

[25] S. Rajendran, S. K. Mathivanan, P. Jayagopal et al., "Emphasizing privacy and security of edge intelligence with machine learning for healthcare," *International Journal of Intelligent Computing and Cybernetics*, 2021, In Press.

[26] N. Best, J. Ott, and E. J. Linstead, "Exploring the efficacy of transfer learning in mining image-based software artifacts," *Journal of Big Data*, vol. 7, no. 1, pp. 1–10, 2020.

[27] E. Gayakwad, J. Prabhu, R. V. Anand, and M. S. Kumar, "Training time reduction in transfer learning for a similar dataset using deep learning," in *Intelligent Data Engineering and Analytics*, pp. 359–367, Springer, Singapore, 2021.

[28] M Venkatasen, S. K Mathivanan, P Jayagopal et al., "Forecasting of the SARS-CoV-2 epidemic in India using SIR model, flatten curve and herd immunity," *Journal of Ambient Intelligence and Humanized Computing*, vol. 15, pp. 1–9, 2020.

[29] A. Lang, A. Carass, M. Hauser et al., "Retinal layer segmentation of macular OCT images using boundary classification," *Biomedical Optics Express*, vol. 4, no. 7, pp. 1133–1152, 2013.

[30] S. Motamedi, K. Gawlik, N. Ayadi et al., "Normative data and minimally detectable change for inner retinal layer thicknesses using a semi-automated OCT image segmentation pipeline," *Frontiers in Neurology*, vol. 10, p. 1117, 2019.

[31] U. Subramaniam, M. M. Subashini, D. Almakhles, A. Karthick, and S. Manoharan, "An expert system for COVID-19 infection tracking in lungs using image processing and deep learning techniques," *BioMed Research International*, vol. 2021, Article ID 1896762, 17 pages, 2021.

[32] N. Tsiknakis, D. Theodoropoulos, G. Manikis et al., "Deep learning for diabetic retinopathy detection and classification based on fundus images: a review," *Computers in Biology and Medicine*, vol. 135, Article ID 104599, 2021.

[33] L. Dai, "A deep learning system for detecting diabetic retinopathy across the disease spectrum," *Nature Communications*, vol. 12, no. 1, pp. 1–11, 2021.
[34] F. Li, H. Chen, Z. Liu et al., "Deep learning-based automated detection of retinal diseases using optical coherence tomography images," *Biomedical Optics Express*, vol. 10, no. 12, pp. 6204–6226, 2019.
[35] L. Huang, X. He, L. Fang, H. Rabbani, and X. Chen, "Automatic classification of retinal optical coherence tomography images with layer guided convolutional neural network," *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 1026–1030, 2019.
[36] N. Rajagopalan, "Deep CNN framework for retinal disease diagnosis using optical coherence tomography images," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–12, 2020.
[37] S. Mahajan, N. Mittal, and A. K. Pandit, "Image segmentation using multilevel thresholding based on type II fuzzy entropy and marine predators algorithm," *Multimedia Tools and Applications*, vol. 80, no. 13, Article ID 19335, 2021.
[38] S. Mahajan, "An efficient adaptive salp swarm algorithm using type II fuzzy entropy for multilevel thresholding image segmentation," *Computational and Mathematical Methods in Medicine*, vol. 2022, Article ID 2794326, 14 pages, 2022.
[39] S. S. Ganesh, G. Kannayeram, A. Karthick, and M. Muhibbullah, "A novel context aware joint segmentation and classification framework for glaucoma detection," *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 2921737, 2021.
[40] S. Mahajan and A. K. Pandit, "Image segmentation and optimization techniques," *A Short Overview*, vol. 2, no. 2, pp. 47–49, 2022.
[41] S. Mahajan, "Hybrid Aquila optimizer with arithmetic optimization algorithm for global optimization tasks," *Soft Computing*, pp. 1–19, 2022.
[42] G. Huang, "Densely Connected Convolutional Networks," p. 1608, 2018, https://arxiv.org/abs/1608.06993.
[43] K. B. Priya, P. Rajendran, S. Kumar et al., "Pediatric and geriatric immunity network mobile computational model for COVID-19," *International Journal of Pervasive Computing and Communications*, 2020, In Press.
[44] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," PMLR, 2015, https://arxiv.org/abs/1502.03167.
[45] C. Szegedy, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, June, 2016.
[46] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, July, 2017.
[47] S. Rajendran and P. Jayagopal, "Accessing Covid19 epidemic outbreak in Tamilnadu and the impact of lockdown through epidemiological models and dynamic systems," *Measurement*, vol. 169, Article ID 108432, 2021.
[48] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2016, https://arxiv.org/abs/1605.07678.
[49] M. S. Kumar and J. Prabhu, "Recent development in big data analytics," *Research Anthology on Artificial Intelligence Applications in Security*, vol. 47, pp. 1640–1663, 2021.
[50] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "The Bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, Citeseer, Orlando Florida, USA, October, 1999.
[51] Z. Lin, S. Mu, A. Shi, C. Pang, and X. Sun, "A novel method of maize leaf disease image identification based on a multichannel convolutional neural network," *Transactions of the ASABE*, vol. 61, no. 5, pp. 1461–1474, 2018.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4587880, 26 pages
https://doi.org/10.1155/2022/4587880

Research Article

K-Means Segmentation of Underwater Image Based on Improved Manta Ray Algorithm

Donglin Zhu,[1,2] **Linpeng Xie**,[1] **and Changjun Zhou**[1]

[1]*College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China*
[2]*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China*

Correspondence should be addressed to Changjun Zhou; zhouchangjun@zjnu.edu.cn

Received 30 December 2021; Revised 3 February 2022; Accepted 8 February 2022; Published 16 March 2022

Academic Editor: Diego Oliva

Image segmentation plays an important role in daily life. The traditional K-means image segmentation has the shortcomings of randomness and is easy to fall into local optimum, which greatly reduces the quality of segmentation. To improve these phenomena, a K-means image segmentation method based on improved manta ray foraging optimization (IMRFO) is proposed. IMRFO uses Lévy flight to improve the flexibility of individual manta rays and then puts forward a random walk learning that prevents the algorithm from falling into the local optimal state. Finally, the learning idea of particle swarm optimization is introduced to enhance the convergence accuracy of the algorithm, which effectively improves the global and local optimization ability of the algorithm simultaneously. With the probability that K-means will fall into local optimum reducing, the optimized K-means hold stronger stability. In the 12 standard test functions, 7 basic algorithms and 4 variant algorithms are compared with IMRFO. The results of the optimization index and statistical test show that IMRFO has better optimization ability. Eight underwater images were selected for the experiment and compared with 11 algorithms. The results show that PSNR, SSIM, and FSIM of IMRFO in each image are better. Meanwhile, the optimized K-means image segmentation performance is better.

1. Introduction

In recent years, image segmentation has attracted much attention and research by researchers. It is of great significance to the future image processing field. As a key step of image processing, image segmentation plays an important role in extracting objects of interest from images. At present, it has important research value in medicine, agriculture, ocean, and other fields. Image segmentation can be divided into four categories: threshold segmentation, region segmentation, edge segmentation, and segmentation methods based on specific theories. The clustering algorithm is a typical unsupervised learning algorithm. It uses the idea of clustering differentiation to solve the problem. The way to solve the problem is simple and easy to understand. It has been successfully applied in many fields [1]. Cluster image segmentation has also been successfully studied. K-means is the most common and easiest clustering method among them, but K-means has the disadvantage of large randomness and easily fall into local optimum, which makes it impossible to control the cluster center reasonably. Swarm intelligence algorithms is an algorithm with global optimization performance and strong versatility that is suitable for parallel processing. This type of algorithm can find the optimal solution or approximate the optimal solution within a certain period of time. [2] Intelligent optimization algorithm opens up a new way for image segmentation. In terms of clustering segmentation, Hrosik *R* C and others improved the K-means clustering algorithm based on the firefly algorithm, which could achieve better segmentation average error, peak signal-to-noise ratio, and structural similarity index on medical images; [3] Li *h* and others proposed a k-means clustering algorithm based on dynamic particle swarm optimization (DPSO), which had better visual effect than traditional K-means clustering in image segmentation and obvious advantages in improving image segmentation quality and efficiency; [4] Shubham and others applied gray wolf optimizer (GWO) [5] to the segmentation of satellite images; [6] Therefore, an intelligent optimization algorithm is of great significance in the field of image segmentation.

With the rapid development of swarm intelligence algorithms, a variety of new algorithms are emerging. In addition to the algorithms mentioned above, there are other algorithms as follows: monarch butterfly optimization (MBO) [7], elephant herding optimization (EHO) [8], moth search (MS) algorithm [9], Harris hawks optimization (HHO) [10], etc. Manta ray foraging optimization (MRFO) is a new swarm intelligence optimization algorithm proposed in 2020. With excellent searchability, fewer parameters, simple model and easily understood, it is better than particle swarm optimization (PSO) [11, 12], genetic algorithm (GA) [13, 14], Differential Evolution (DE) [15, 16], Cuckoo Search (CS) [17], gravitational search algorithm (GSA) [18], and ABC [19] in some function optimization [20], and it has been successfully applied to solar energy [21, 22], ECG [23], generator [24, 25], power system [26], cogeneration energy system [27], geophysical inversion problem [28], directional overcurrent relay [29], feature selection [30], hybrid energy system [31], and sewage treatment [32]. Although MRFO has good optimization ability, it still has its own defects. In complex problems, the search ability is limited and the diversity of the population can not be guaranteed. The main reason is that the individual searches orderly, highly dependently, and inflexibly.

At present, researchers have noticed this point and carried out successive studies, such as Mohamed Abd Elaziz, who combines fractional calculus with MRFO to correct the direction of manta ray movement. This algorithm has been verified by CEC 2017 test function and is applied to image segmentation problems with good feasibility [33]. Mohamed H. Hassan combines a gradient optimizer with MRFO to reduce the probability that the algorithm will fall into a local optimum, which has a good effect in single-objective and multiobjective economic emission scheduling [34]. Haitao Xu uses adaptive weighting and chaos to improve MRFO, so as to efficiently handle thermodynamic problems [35]. Essam H. Houssein uses reverse learning to initialize the population so as to increase the diversity of the population and apply it to the threshold image segmentation problem with good segmentation quality [36]. Bibekananda Jena adds an attack capability to MRFO, which allows it to jump out of local optimization and find a globally optimal solution. It is then applied to the image segmentation problem of 3D Tsallis [37]. Mihailo Micev fuses Simulate Anneal (SA) with MRFO and applies it to the Proportional Integral Derivative (PID) controller. The fused algorithm is superior to other algorithms [38]. In addition, Serdar and others adopt opposition-based learning and SA to improve the convergence effect of MRFO. It has better control performance when applied to fractional order proportional integral derivative (FOPID) controller [39]. Although the currently proposed variants of MRFO have achieved some results, the following problems still exist:

(1) Most scholars use the fusion of other algorithms to improve the search ability, but this will bring higher time complexity, and the algorithm after fusion may not be able to complement each other so as to present perfect results.

(2) Reverse learning can only solve inversely in a certain space, but in complex high-dimensional situations, there are fewer individual optimization methods, and they cannot jump out of the local optimal state perfectly.

(3) In the optimization process, the above algorithm cannot completely balance the local search and global search capabilities, which results in insufficient convergence accuracy of the algorithm.

Based on the above analysis, this paper presents an improved algorithm for manta rays, which uses random walk learning to make individuals wander randomly in space, to increase the diversity of the population, and avoid premature convergence of the algorithm, and then we use Lévy flight for long-distance and short-distance searches to balance the local and global searches of the algorithm. Finally, the learning idea of particle swarm is introduced. Two learning factors are used to improve the convergence accuracy of the algorithm 12 functions are used to verify the validity and feasibility of IMRFO. Then eight underwater image datasets are used in K-means image segmentation. The results show that IMRFO has better generalization ability and better segmentation quality.

The innovations and contributions of this paper are as follows:

(i) A random walk learning algorithm is designed to increase the diversity of the population and reduce the probability of the algorithm falling into local optimization to a certain extent.

(ii) Lévy flight and learning factors are introduced to balance the searchability of the algorithm, which makes the algorithm have a good convergence effect.

(iii) In 12 standard test functions, IMRFO is compared with 7 other algorithms to show its superiority and feasibility. Next, two statistical tests are used to emphasize the optimization performance of the algorithm. It is compared with the recently proposed variants of the algorithm. Finally, ablation experiments were performed, all the results show that IMRFO has a good search ability.

(iv) IMRFO is applied to K-means underwater image segmentation. The results of 11 algorithms show that IMRFO performs well.

The structure of this paper is as follows: Section 2 introduces the basic MRFO algorithm, Section 3 introduces the improved IMRFO algorithm and related analysis. Section 4 describes the process of IMRFO optimizing K-means image segmentation. Section 5 tests the performance of IMRFO and compares the related algorithms. Section 6 describes and analyses the performance of each algorithm in K-means image segmentation. Section 7 summarizes the experimental results of this paper. The last section expresses the advantages and disadvantages of IMRFO and future research directions.

2. Manta Ray Foraging Optimization

Manta rays feed on plankton, which are mainly water microfauna. When feeding, they suck water and prey into their mouths with angular head leaves. They then filter prey out of the water through improved rabbles. Individuals of the manta rays work together to find the best food. Inspired by the behavior of the manta rays, the algorithm is divided into chain feeding, spiral-feeding, and somersault foraging. There are three stages of spiral and empty foraging.

2.1. Chain Feeding. At this stage, the manta ray population will be arranged in an ordered chain to collaborate in feeding, which will maximize the amount of plankton in the pocket. The mathematical model of the chain feeding process can be expressed as follows:

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 1, \\ x_i^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 2, 3, \ldots, N. \end{cases} \tag{1}$$

In formula (1), $x_i^d(t)$ denotes the d-dimensional information of the location of the first manta ray in generation t. R is a random number that obeys a uniform distribution of [0,1]. $\alpha = 2 \cdot r \cdot \sqrt{|\log(r)|}$ is the weight factor, $x_{\text{best}}^d(t)$ is the d-dimensional information of the best location found so far The manta ray at position i depends on the manta ray at position i-1 and the best food position found so far. N represents the population number. The update of the first manta ray depends on the optimal location.

2.2. Spiral Feeding. When a manta ray finds a good food source in a certain space, each individual approaches a manta ray in front of it, in addition to spirally moving toward the food. The spiral-feeding process can be represented by the following mathematical model:

$$x_i^d(t+1) = \begin{cases} x_{best}^d(t) + r \cdot \left(x_{best}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{best}^d(t) - x_i^d(t)\right), & i = 1, \\ x_{best}^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{best}^d(t) - x_i^d(t)\right), & i = 2, 3, \ldots, N. \end{cases} \tag{2}$$

where $\beta = 2e^{r_1(T-t+1)/T} \cdot \sin(2\pi r_1)$, a weight factor representing the spiral motion, T being the largest number of iterations, r being the rotation factor and obeying [0,1] uniform random numbers. In addition, in order to improve the efficiency of population foraging, MRFO randomly generates a new location during the optimization process and then performs a spiral search at that location. Its mathematical model is as follows:

$$x_i^d(t+1) = \begin{cases} x_{\text{rand}}^d(t) + r \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 1, \\ x_{\text{rand}}^d(t) + r \cdot \left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta \cdot \left(x_{\text{best}}^d(t) - x_i^d(t)\right), & i = 2, 3, \ldots, N. \end{cases} \tag{3}$$

$x_{\text{rand}}^d(t)$ represents a new location in space.

2.3. Somersault Foraging. When a certain manta ray finds a food source, its position can be regarded as an axis. Each manta ray tends to wander around the axis and flip to a new location. Its mathematical model is as follows:

$$x_i^d(t+1) = x_i^d(t) + S \cdot \left(r_2 \cdot x_{\text{best}}^d(t) - r_3 \cdot x_i^d(t)\right), i = 1, 2, \ldots, N. \tag{4}$$

S is the flip factor, which determines the flip distance. R_2 and r_3 are two random numbers that are uniformly distributed [0,1]. As S values vary, individual mantas flip to locations in search space that are symmetrical to the optimal solution at their current location.

3. Improved Manta Ray Foraging Optimization

From the above formulas, it can be seen that more communication between individuals and orderly work can improve the searchability of the algorithm and perform a wide-ranging search. On the one hand, the lack of initiative of individuals in the population limits their ability to develop. On the other hand, updates within the population are related to the best location. When encountering high-dimensional complex problems, the change of the optimal position is similar, which results in less change in the two updates before and after the algorithm, which limits the algorithm's optimization ability. Therefore, a flexible change strategy is needed to improve the development ability and local convergence effect of the algorithm. This paper uses the

Lévy flight strategy to improve individual blindness search, and random walk learning is used to prevent the algorithm from falling into a local state and the learning idea of particle swarm to improve the search accuracy of the algorithm.

3.1. Why Each Modification Has Been Proposed? MRFO is based on a group of animals collaborating in feeding, which results in fewer optimization methods and a lack of flexibility and fineness. Therefore, individual initiative is required to increase the diversity of the population in order to find high-quality solutions in space.

Therefore, this paper analyses and solves the defects of the algorithm from the following three points. Firstly, it is necessary to make the population individuals better distribute the whole space so as to develop the vision of the algorithm and improve the global search ability of the algorithm. Lévy flight is a classical strategy, which can fly in a given space in the way of alternating long and short distances. It has been used by most scholars to improve the search ability of the algorithm. Secondly, some individuals need to be independent and never be limited by group characteristics. Random walk learning is an uncertain way of walking. The traditional random walk can only be carried out in local areas. However, the random walk learning designed in this paper can make large location differences between different individuals and improve the population diversity of the algorithm. Finally, information sharing among individuals is needed to improve the local search ability of the algorithm and find high-quality solutions. The learning factor is derived from the particle swarm optimization algorithm, which is used to speed up the information exchange of the population, prevent the early invalid search, improve the local search ability of the algorithm, and improve the accuracy of the solution to a certain extent.

3.2. Lévy Flight Strategy. When manta ray individuals perform chain search, all individuals follow the population to search, which leads to the lack of flexibility of the algorithm and can not perform a better search range. Therefore, the Lévy flight strategy [40, 41] is introduced to enable individuals to search long and short distances, increase the diversity of the population, and enable individuals to fully diffuse into the whole space. The location update format for joining Lévy flight strategy is as follows:

$$x_i'(t) = x_i(t) + l \oplus \text{levy}(\lambda). \tag{5}$$

In formula (8), $x_i(t)$ represents the position of the i-th individual in the t-th iteration, $\oplus$ is an arithmetic symbol representing point-to-point multiplication. $l = 0.01(x_i(t)\text{-}x_p)$ denotes a step length control parameter, x_p represents the position of the best individual in the population.

Lévy flight formula is as follows [42]:

$$\text{Levy}(x) = 0.01 \times \frac{r_4 \times \sigma}{|r_5|^{(1/\xi)}}. \tag{6}$$

where r_4 and r_5 are random numbers within the range of [0,1], ξ. The general value is 1.5. σ is calculated as follows:

$$\sigma = \left(\frac{\Gamma(1+\xi) \times \sin(\pi\xi/2)}{\Gamma(1+\xi/2) \times \xi \times 2^{(\xi-1/2)}} \right)^{(1/\xi)}. \tag{7}$$

where $\Gamma(x) = (x-1)!$, the schematic diagram of Lévy flight is shown in Figure 1. Lévy flight can search long and short distances in a certain space and balance the global and local search of the algorithm.

3.3. Random Walk Learning. In the optimization process, MRFO has the probability of falling into the local optimum, which makes the current optimum individual unreliable, so it is necessary to disperse all the individuals to find a better solution. Unlike random walks, the learning factors at the best and worst locations are introduced to make individual escape directional and reduce unreasonable walk. The specific mathematical model of RWL is as follows:

$$x_{i,j}^{t+1} = x_{\text{best}}^t + \left(\frac{2}{1 + t/M \cdot \sin(\pi/2 \cdot r)} - 1 \right) \cdot e^{\left(-\frac{t}{M}\right) \cdot \left(x_{i,j}^t - x_{\text{worst}}^t\right)} + \left(c_1 \cdot x_{\text{best}}^t - c_2 \cdot X_{i,j}^t\right). \tag{8}$$

In formula (8), $((2/1 + t/M.\sin(\pi/2.r)) - 1)$ is the sinusoidal random factor that uses the mathematical properties of the sinusoidal function to fluctuate toward the optimal solution and continuously adjusts the step size based on the worst position of the current population so that the search path can span the entire solution space. M is the maximum number of iterations, and c_1 and c_2 represent two learning factors, random numbers that obey a normal distribution. $e^{(-t/M).(x_{i,j}^t - x_{\text{worst}}^t)}$ is the control step, $(c_1.x_{\text{best}}^t - c_2.X_{i,j}^t)$ is the direction of control. As shown in Figure 2(a), (a) is the distribution of individuals without introducing RWL, (b) introduces the individual distribution of RWL; we can see that the introduction of RWL enables individuals to master global information, makes the individual distribution more even, and finds the global optimal solution.

3.4. PSO Algorithm Learning Ideas. There are two learning factors in PSO to develop local solutions, which can effectively improve the convergence accuracy of the algorithm. Therefore, the formula of introducing two learning factors is as follows:

$$x_{i,j}^{t+1} = b_1 \cdot x_{i,j}^t + b_2 \cdot \text{rand}(1) \cdot \left(\text{Best}X - x_{i,j}^t\right). \tag{9}$$

b_1, b_2 are two learning factors, and *BestX* is the optimal position of the current population. As can be seen from the formula, this strategy exploits individuals between the current one and the optimal one to enhance the local search of the algorithm.

3.5. Improved Manta Ray Foraging Optimization. To improve the local search capability of MRFO and reduce the probability of falling into local optimum, an improved manta ray algorithm is presented in this paper. The

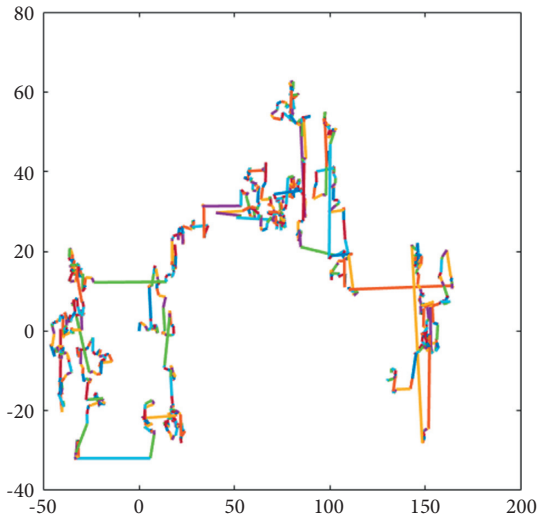


Figure 1: Lévy flight diagram.

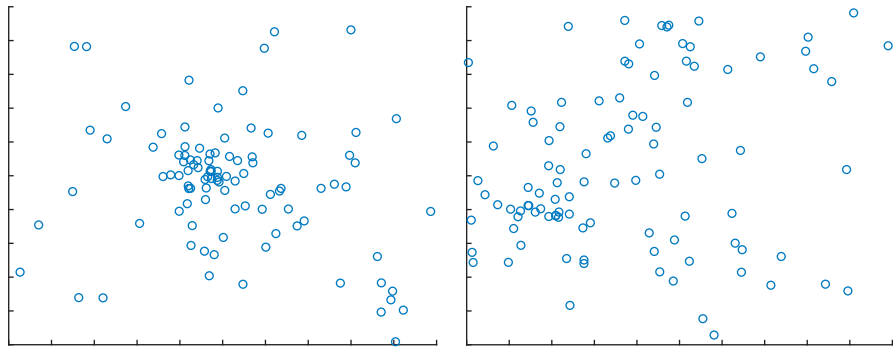


Figure 2: Distribution of algorithm individuals (a) original algorithm (b) algorithm with GWL.

algorithm uses random walk learning to prevent the algorithm from falling into the local state after each iteration and to improve the global search ability of the algorithm. Then, the Lévy flight mechanism is combined to improve the blindness of the manta ray algorithm and to balance the searchability of the algorithm. Using two learning factors of particle swarm optimization to improve the search accuracy ultimately makes the algorithm improve effectively both in local and global aspects. The specific pseudocode is shown in Algorithm 1.

4. K-Means Image Segmentation Based on IMRFO

The principle of the traditional K-means algorithm is to select K cluster centers randomly, so the way to select them is uncertain, resulting in large differences in the final results and easy to fall into local optimum. Therefore, it is necessary to select an appropriate initial cluster center. Intelligent optimization algorithm has been successfully applied to K-means to improve its randomness and the defect of falling into local optimum. The improved manta ray foraging optimization optimizes K-means so that the initial cluster centers are well controlled. The objective function is as follows:

$$f = \sum_{i=1}^{n} \sum_{j=1}^{m} \left| X_i - Y_i \right|^2. \tag{10}$$

X_i is a pixel gray value of the image and Y_i is the J-th clustering center. The optimal initial number of clustering centers is obtained by IMRFO to minimize the fitness value of the objective function.

K-means image segmentation based on IMRFO is mainly divided into two parts:

(1) Use the global search capabilities of IMRFO to find the best initial cluster center in the image point set

(2) The initial cluster centers of the IMRFO output are segmented in the K-means algorithm

The specific flow chart is shown in Figure 3.

5. Performance Analysis and Test

5.1. Performance Test. To verify the effectiveness and feasibility of IMRFO, 12 benchmark functions [43, 44] are

Input
M : Maximum number of iterations
N : Population sparrows
Output : X_{best}, f_g
Initialize population
$t = 1$;
While($t < M$)
Update the position of the population according to formula (8)%RWL
For $i = 1$ to N do
If rand<0.5 then
If t/M < rand then
Update the position of the individual according to formula (3)
Else
Update the position of the individual according to formula (2)
End if
Else
Update the position of the individual according to formula (1)
End if
Update the position of the population according to formula (5)% Lévy flight
Get the current best and worst
For $i = 1$ to N do
Update the position of the individual according to formula (4)
End for
The position of the individual is updated according to formula (9)% PSO algorithm learning ideas
Get a new Xbest, fg, X_{worst};
End while
Get the final Xbest, fg.

Algorithm 1: The framework of the IMRFO.

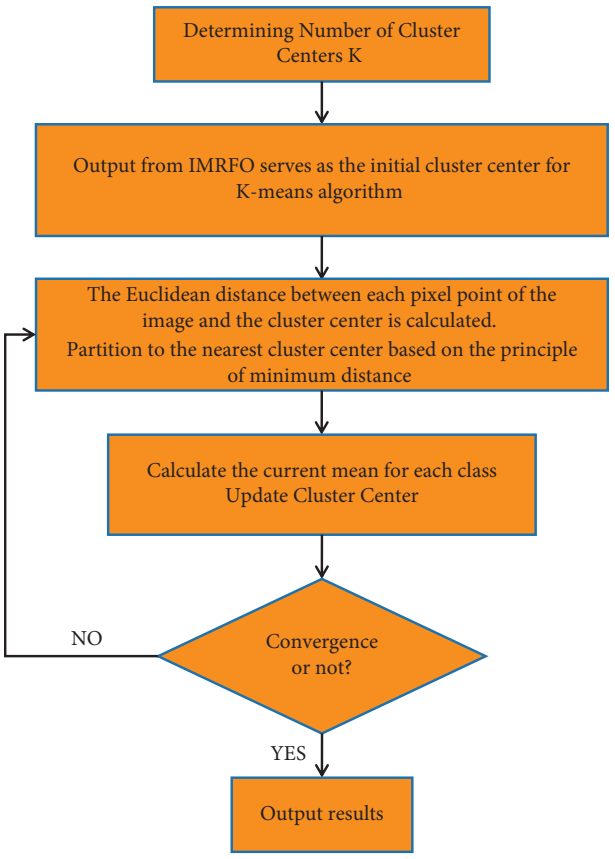


Figure 3: K-Means image segmentation based on IMRFO.

selected to verify its function optimization ability. The specific test function information is shown in Table 1. F1-6 is a unimodal function, F7-11 is a complex multimodal function, and F12 is a fixed-dimensional function. In addition, F1-11 is tested in different dimensions to verify the optimization ability of the algorithm in high-dimensional cases. To prove that IMRFO is competitive, seven algorithms including MRFO, Honey Badger Algorithm (HBA) [45], GWO, PSO, Whale Optimization Algorithm (WOA) [46], Learning Based Optimization (TLBO) [47], and Flower Pollution Algorithm (FPA) [48] are compared. The new cluster intelligence algorithm was proposed by HBA in 2021, while other algorithms are classical ones that have been extensively studied. The number of iterations and population of each algorithm are 500 and 100. In HBA, $O = 6$, $C = 2$; In FPA, the selection probability $p = 0.8$. B_1 and b_2 in IMRFO are 0.2 and 0.8, respectively. The experimental environment is Windows 10 64 bit; the software is matlab2019b; the memory is 16 GB; the processor is Intel(R) Core (TM) i5-10200H CPU @ 2.40 GHz. The average, optimal value, and standard deviation of the results of each algorithm for 30 runs are calculated. If IMRFO is the optimal value, the font is bolded. The optimization results of each algorithm are calculated as shown in Table 2-3.

On the one hand, from Table 2 and Table 3, we can see that IMRFO has obvious advantages in searching ability, and the results are better than other algorithms in each function. The increase of dimension does not reduce the searching ability of IMRFO. On the other hand, among these functions, F1, F6, F8-10, F12, MRFO itself has a good optimization effect and can find the theoretical optimal value, IMRFO also has the same optimization effect, so it can be seen that IMRFO does not weaken the original algorithm's optimization ability. Overall, IMRFO has been effectively improved in stability and accuracy. It can be seen that the introduction of multiple strategies improves the algorithm's optimization ability and reduces the probability of entering a local optimum.

5.2. Statistical Test. To verify whether IMRFO and the other seven algorithms have significant differences in global optimization, the 30-dimension results of each algorithm are tested. The Wilcoxon rank-sum test is used to find the differences between the two algorithms. Assume H0 : The two algorithms have the same performance. H1 : There is an obvious difference between the two algorithms. Use the P-value of the test results to measure the differences between the two algorithms. When $P < 0.05$, reject H0. It shows that there is a significant difference between the two algorithms. When $P > 0.05$, H0 is accepted, indicating that the two algorithms have the same global optimization performance. To clearly see the differences between these algorithms, we utilize N/A to represent the values of $P > 0.05$. The Wilcoxon test results are shown in Table 4. At the same time, in order to better show the comprehensive optimization ability of IMRFO in the whole test function, the average and variance of each algorithm are Friedman test [49], and the final ranking is calculated to measure the universality of the algorithm in the 12 test functions. The test results are shown in Table 5.

From Table 4, it can be seen that IMRFO differs significantly from other algorithms. In some functions, MRFO itself has better searching ability, so the difference is not obvious. From Table 5, IMRFO ranks best in the search results of each function, which also indicates that it has a good universality.

5.3. Comparison with Variants of the Algorithm. To further show the effectiveness and innovation of IMRFO, this paper compares IMRFO with multistrategy serial cuckoo search algorithm (MSSCS) [50], firefly algorithm with courtship learning (FACL) [51], self-adaptive cuckoo search (SACS) [52], and CSsin [53] proposed in recent years. These four algorithms are variants of classical algorithms and have been validated in the CEC test set. The specific parameters of each algorithm are set as follows: In MSCS algorithm, $\alpha = 0.01$, $\beta = 1.5$, $P_a = 0.25$, $C = 0.2$, $PA_{max} = 0.35$, $PA_{min} = 0.25$; in CSsin algorithm, $P_{\max} = 0.75$, $P_{\max} = 0.25$, $freq = 0.5$; in FACL, $\alpha = 0.01$, $\beta_{\min} = 0.2$, $\beta = 1$, $\gamma = 1$. The number of populations and the number of iterations for each algorithm are shown above. Similarly, when IMRFO is the optimal value, font bolding will be applied. The results of each algorithm are shown in Table 6.

From Table 6, it is clear that IMRFO is the best value in F1-4, F6, F9-12, which shows that IMRFO is better than these algorithms in the optimization of these functions. Secondly, the variants of CS have better optimization results, especially in F5 and F7, which have higher accuracy. FACL, as the worst one, has poor optimization results but good stability. Generally speaking, IMRFO has some advantages in function optimization, which verifies the effectiveness and innovation of the algorithm.

5.4. Convergence Analysis. In order to clearly see the optimization and convergence effect of each algorithm in each function, the average convergence diagram of each algorithm is given as shown in Figure 4.

From Figure 4, it can be seen that IMRFO has a good convergence effect and can find the most accurate solution quickly, especially in the functions of F1-4, F6, F11. It can be seen that the flexible search mechanism enables the algorithm to find the best solution quickly in the optimization process.

5.5. Ablation Experiment. In order to verify the validity and feasibility of the three combinations of strategies, the combinations of strategies are experimented with to find the better one. In this paper, the algorithm of combining Lévy flight with GWL is recorded as MRFO-I, the algorithm of combining Lévy flight with PSO learning thought strategy is recorded as MRFO-II, while the algorithm of combining PSO thought with GWL is recorded as MRFO-III. Besides, the algorithm using Lévy flight alone is recorded as MRFO-IV, the algorithm using GWL alone as MRFO-V, and the algorithm utilizing PSO alone as MRFO-VI. The

Table 1: Test function information table.

F	Dim	Interval	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/100	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} x_i$	30/100	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30/100	[-100,100]	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30/100	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30/100	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30/100	[-1.28,1.28]	0
$F_7(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30/100	[-500,500]	-418.9829n
$F_8(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30/100	[-5.12,5.12]	0
$F_9(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	30/100	[-32,32]	0
$F_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30/100	[-600,600]	0
$F_{11}(x) = (\pi/n)\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + (x_i + 1/4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a < x_i < a, \\ k(-x_i - a)^m & x_i < -a. \end{cases}$	30/100	$[-50, 50]$	0
$F_{12}(x) = (0.002 + \sum_{i=1}^{25}(1/i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6))^{-1}$ where $a = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$	2	[-65.536.65.536]	0.998

Table 2: Performance comparison table of each algorithm

F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	0	0	**0**
	MRFO	0	0	0
	HBA	$9.3799E-166$	$1.1587E-160$	$4.2983E-160$
	WOA	$3.1181E-104$	$1.0841E-95$	$4.5034E-95$
	TLBO	$4.0530E-86$	$2.9876E-85$	$2.1878E-85$
	FPA	$3.4196E+04$	$5.6323E+04$	$9.3090E+03$
	GWO	$2.97266E-42$	$9.97975E-41$	$2.31134E-40$
	PSO	$1.35628E-12$	$7.30343E-11$	$1.53846E-10$
$F_2(x)$	IMRFO	**0**	**$4.8161E-277$**	**0**
	MRFO	$4.2592E-237$	$1.5644E-229$	0
	HBA	$1.0045E-87$	$3.0642E-85$	$8.9871E-85$
	WOA	$5.1550E-63$	$8.1673E-58$	$2.9729E-57$
	TLBO	$3.0778E-43$	$1.2493E-42$	$7.1358E-43$
	FPA	$4.4336E+07$	$7.9783E+10$	$1.3371E+11$
	GWO	$1.00188E-24$	$5.16621E-24$	$3.79647E-24$
	PSO	$1.9394E-06$	$1.2453E-04$	$3.7943E-04$
$F_3(X)$	IMRFO	**0**	**0**	**0**
	MRFO	0	$4.2858E-192$	0
	HBA	$7.5465E-129$	$1.8442E-117$	$6.9103E-117$
	WOA	$4.7217E+03$	$1.4248E+04$	$6.9878E+03$
	TLBO	$6.0458E-17$	$1.7598E-15$	$1.8686E-15$
	FPA	$6.4669E+04$	$9.4629E+04$	$1.9121E+04$
	GWO	$1.2723E-15$	$1.6522E-11$	$5.6380E-11$
	PSO	2.4597	6.8764	4.6664
$F_4(X)$	IMRFO	**0**	**$3.6613E-283$**	**0**
	MRFO	$1.8994E-233$	$1.2937E-223$	0
	HBA	$6.4810E-72$	$3.6559E-68$	$8.7750E-68$
	WOA	$1.3789E-07$	$2.9967E+01$	$3.0998E+01$
	TLBO	$7.54383E-35$	$2.2303E-34$	$1.1471E-34$
	FPA	$7.5982E+01$	$8.3319E+01$	$3.0315E+00$
	GWO	$1.0249E-11$	$1.3528E-10$	$1.3630E-10$
	PSO	$1.0009E+01$	$1.0221E+02$	$5.1654E+01$
$F_5(X)$	IMRFO	$1.8761E+01$	$1.9089E+01$	$4.2393E-01$
	MRFO	$1.9091E+01$	$2.0074E+01$	$3.9236E-01$
	HBA	$1.9310E+01$	$2.1199E+01$	$7.6666E-01$
	WOA	$2.6313E+01$	$2.6797E+01$	$2.0942E-01$
	TLBO	$1.6702E+01$	$1.8846E+01$	1.0315
	FPA	$1.0583E+08$	$1.7478E+08$	$3.8136E+07$
	GWO	$4.5186E+01$	$4.6672E+01$	$7.8488E-01$
	PSO	89.4507	234.8606	104.6002
$F_6(X)$	IMRFO	**$1.4166E-06$**	**$2.9500E-05$**	**$2.0356E-05$**
	MRFO	$3.1973E-06$	$3.8448E-05$	$4.0550E-05$
	HBA	$1.6931E-05$	$1.1151E-04$	$8.9566E-05$
	WOA	$3.9351E-05$	$9.5517E-04$	$9.4421E-04$
	TLBO	$4.4041E-04$	$7.9732E-04$	$2.1545E-04$
	FPA	$5.1790E+01$	$9.3728E+01$	$1.9183E+01$
	GWO	$1.8171E-04$	$5.6482E-04$	$3.1680E-04$
	PSO	$7.1930E-03$	$1.8163E-02$	$7.2361E-03$
$F_7(X)$	IMRFO	$-1.0536E+04$	$-1.0050E+04$	$4.2525E+02$
	MRFO	$-9.6874E+03$	$-9.0741E+03$	$4.3594E+02$
	HBA	$-1.0990E+04$	$-9.4427E+03$	$9.4018E+02$
	WOA	-12569.4699	-11724.03485	1302.326078
	TLBO	$-9.2643E+03$	$-7.3757E+03$	$1.1529E+03$
	FPA	$-4.0064E+03$	$-2.9596E+03$	$3.8588E+02$
	GWO	$-8.2622E+03$	$-6.3621E+03$	$7.0903E+02$
	PSO	$-8.3241E+03$	$-6.8450E+03$	$7.5268E+02$

Table 2: Continued.

F	Algorithm	Best	Mean	Std
$F_8(X)$	IMRFO	**0**	**0**	**0**
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	$1.8948E-15$	$1.0378E-14$
	TLBO	0	6.3284	4.7427
	FPA	$3.3516E+02$	$3.9820E+02$	$2.9755E+01$
	GWO	0	$4.7528E-01$	1.4588
	PSO	0	2.0630	1.3861
$F_9(X)$	IMRFO	**$8.8818E-16$**	**$8.8818E-16$**	**0**
	MRFO	$8.8818E-16$	$8.8818E-16$	0
	HBA	$8.8818E-16$	$8.8818E-16$	0
	WOA	$8.8818E-16$	$4.0856E-15$	$2.6960E-15$
	TLBO	$4.4409E-15$	$6.3357E-15$	0
	FPA	$1.9290E+01$	$2.0337E+01$	$2.7610E-01$
	GWO	$1.8652E-14$	$2.7652E-14$	$3.8165E-15$
	PSO	$1.7249E-06$	$1.9663E-01$	$4.5356E-01$
$F_{10}(X)$	IMRFO	**0**	**0**	**0**
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	$1.9723E-03$	$3.3266E-03$
	TLBO	0	0	0
	FPA	$2.7736E-01$	$1.2829E+00$	$6.1301E-01$
	GWO	0	$2.4654E-03$	$3.5462E-03$
	PSO	0	$1.9723E-03$	$3.3266E-03$
$F_{11}(X)$	IMRFO	**$7.3400E-20$**	**$1.2392E-18$**	**$1.2894E-18$**
	MRFO	$4.04531E-06$	$1.53518E-05$	$8.77191E-06$
	HBA	$3.8903E-10$	$6.7898E-09$	$1.1908E-08$
	WOA	$2.7893E-03$	$4.9054E-03$	$1.5734E-03$
	TLBO	$5.7527E-05$	$2.2424E-03$	$5.7778E-03$
	FPA	$5.1790E+07$	$3.7356E+08$	$1.3609E+08$
	GWO	$1.4152E-06$	$1.4681E-02$	$9.2775E-03$
	PSO	$6.4838E-14$	$8.6436E-02$	$2.5276E-01$
$F_{12}(X)$	IMRFO	**$9.9800E-01$**	**$9.9800E-01$**	**0**
	MRFO	$9.9800E-01$	$9.9800E-01$	0
	HBA	$9.9800E-01$	$9.9800E-01$	0
	WOA	$9.9800E-01$	1.2626	$6.8599E-01$
	TLBO	$9.9800E-01$	$9.9800E-01$	0
	FPA	1.9995	$1.0262E+01$	7.6364
	GWO	$9.9800E-01$	$1.7255E+00$	$9.7247E-01$
	PSO	$9.9800E-01$	$1.1305E+00$	$3.4368E-01$

The optimal value is shown in bold.

Table 3: Table of results for each algorithm (100 DIMENSIONS).

F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	**0**	**0**	**0**
	MRFO	0	0	0
	HBA	8.9040E-147	4.6592E-141	1.2550E-140
	WOA	1.0469E-102	7.2231E-93	2.5499E-92
	TLBO	1.6114E-75	1.6114E-75	1.1585E-75
	FPA	1.3848 E+05	2.2811 E+05	4.4746 E+04
	GWO	5.0821E-18	2.6308E-17	2.1090E-17
	PSO	1.0669	1.2459 E+01	3.0029 E+01
$F_2(x)$	IMRFO	**2.0908E-305**	**4.0935E-281**	**0**
	MRFO	8.2591E-234	8.7997E-228	0
	HBA	3.2693E-78	1.4848E-75	2.1169E-75
	WOA	3.97136E-62	8.1673E-58	4.04975E-55
	TLBO	1.58996E-38	1.11006E-55	1.64669E-38
	FPA	2.5777 E+27	4.8007 E+45	1.3669 E+46
	GWO	1.1188E-23	1.0071E-22	8.1329E-23
	PSO	1.6825E-06	4.0865E-05	1.3518E-04

Table 3: Continued.

F	Algorithm	Best	Mean	Std
$F_3(X)$	IMRFO	**0**	**0**	0
	MRFO	0	4.2858E-192	0
	HBA	5.8574E-106	6.5344E-94	3.2931E-93
	WOA	4.5880 E+05	6.7389 E+05	1.1901E+05
	TLBO	5.7047E-06	1.4581E-04	1.5072E-04
	FPA	5.9828 E+05	9.8255 E+05	2.2421 E+05
	GWO	4.3996E-01	1.1814 E+01	1.3585 E+01
	PSO	4.1855 E+03	6.8320 E+03	7.6320 E+03
$F_4(X)$	IMRFO	**2.0908E-305**	**4.0728E-281**	**0**
	MRFO	1.0891E-224	6.0212E-219	0
	HBA	4.1580E-49	6.4459E-47	1.2824E-46
	WOA	9.9712E-04	7.4068 E+01	2.9309 E+01
	TLBO	2.64171E-30	4.63468E-30	1.35657E-30
	FPA	8.7604 E+01	9.3681 E+01	2.0880
	GWO	2.0983E-03	5.2094E-02	8.6554E-02
	PSO	7.3783	9.5754	1.2668
$F_5(X)$	IMRFO	**9.0554 E+01**	9.1195 E+01	7.3432E-01
	MRFO	9.0823 E+01	9.2361 E+01	6.6140E-01
	HBA	9.1619 E+01	9.4762 E+01	1.8652 E+00
	WOA	9.6679 E+01	9.7258 E+01	3.0684E-01
	TLBO	9.1198 E+01	9.3064 E+01	9.5488E-01
	FPA	4.0625 E+08	9.3161 E+08	2.3879 E+08
	GWO	9.5679 E+01	9.6973 E+01	9.2081E-01
	PSO	6.5109 E+02	3.2217 E+03	6.7489 E+03
$F_6(X)$	IMRFO	**1.4328E-06**	**6.6842E-05**	4.9030E-05
	MRFO	2.0781E-06	4.9605E-05	4.3316E-05
	HBA	2.3868E-05	1.4383E-04	1.2534E-04
	WOA	8.5089E-06	1.3208E-03	1.3226E-03
	TLBO	5.1751E-04	1.1234E-03	3.2954E-04
	FPA	6.1163 E+02	1.4259 E+03	3.5171 E+02
	GWO	9.4824E-04	2.6521E-03	1.1372E-03
	PSO	4.9985E-01	1.1220 E+00	3.6320E-01
$F_7(X)$	IMRFO	-2.92 E+04	-2.57 E+04	1.67 E+03
	MRFO	-2.7791 E+04	-2.5142 E+04	1.1769 E+03
	HBA	-3.2530 E+04	-2.6092 E+04	3.5550 E+03
	WOA	-4.1898 E+04	-3.9015 E+04	3.7697 E+03
	TLBO	-2.7940 E+04	-1.6598 E+04	5.1640 E+03
	FPA	-6.5597 E+03	-5.3592 E+03	6.0949 E+02
	GWO	-2.2404 E+04	-1.6607 E+04	2.4372 E+03
	PSO	-2.3436 E+04	-2.0057 E+04	2.2771 E+03
$F_8(X)$	IMRFO	**0**	**0**	0
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	3.7896E-15	2.0756E-14
	TLBO	0	6.3284	4.7427
	FPA	1.3434 E+03	1.5311 E+03	8.6281 E+01
	GWO	0	1.3971	2.2329
	PSO	2.8854 E+01	4.8456 E+01	1.1153 E+01
$F_9(X)$	IMRFO	**8.8818E-16**	**8.8818E-16**	**0**
	MRFO	8.8818E-16	8.8818E-16	0
	HBA	8.8818E-16	8.8818E-16	0
	WOA	8.8818E-16	4.3225E-15	2.8731E-15
	TLBO	7.9936E-15	7.9936E-15	0
	FPA	1.9841 E+01	2.0520 E+01	3.3562E-01
	GWO	2.2204E-14	3.0257E-14	3.7242E-15
	PSO	1.3320E-06	1.4424E-01	4.4254E-01

Table 3: Continued.

F	Algorithm	Best	Mean	Std
$F_{10}(X)$	IMRFO	**0**	**0**	0
	MRFO	0	0	0
	HBA	0	0	0
	WOA	0	0	0
	TLBO	0	0	0
	FPA	1.3815 E+03	2.0376 E+03	3.4470 E+02
	GWO	0	1.8958E-03	3.1773E-03
	PSO	0	1.3192E-03	2.8042E-03
$F_{11}(X)$	IMRFO	**2.8193E-06**	**6.4940E-06**	**3.2213E-06**
	MRFO	4.6460E-06	1.09637E-05	6.3630E-06
	HBA	1.9642E-02	3.5856E-02	1.0354E-02
	WOA	1.8029E-03	7.0382E-03	9.1277E-03
	TLBO	7.3698E-05	2.3727E-03	5.8937E-03
	FPA	7.4824 E+08	2.1393 E+09	7.0466 E+08
	GWO	2.1287E-06	1.5932E-02	9.1126E-03
	PSO	2.3297E-13	8.2984E-02	1.7745E-01
$F_{12}(X)$	IMRFO	**9.9800E-01**	**9.9800E-01**	**0**
	MRFO	9.9800E-01	9.9800E-01	0
	HBA	9.9800E-01	9.9800E-01	0
	WOA	9.9800E-01	1.2626	6.8599E-01
	TLBO	9.9800E-01	9.9800E-01	0
	FPA	1.9995	1.0262 E+01	7.6364
	GWO	9.9800E-01	1.7255 E+00	9.7247E-01
	PSO	9.9800E-01	1.1305 E+00	3.4368E-01

Bold font is to clearly see the advantages of the algorithm. The best indicators of IMRFO have been bold in this paper.

Table 4: Wilcoxon rank-sum test p-value.

F	MRFO	FPA	GWO	HBA	PSO	TLBO	WOA
F1	N/A	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F2	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F3	N/A	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F4	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F5	1.46E-09	1.79E-11	1.79E-11	4.94E-11	3.53E-10	N/A	1.79E-11
F6	N/A	2.91E-11	2.91E-11	5.07E-07	2.91E-11	2.91E-11	2.29E-10
F7	7.61E-09	1.44E-11	1.44E-11	3.31E-03	1.44E-11	1.97E-11	7.40E-07
F8	N/A	1.21E-12	8.15E-02	N/A	4.24E-12	1.66E-11	3.34E-01
F9	N/A	1.21E-12	6.59E-13	N/A	1.21E-12	4.63E-13	1.02E-07
F10	N/A	1.21E-12	6.62E-04	N/A	1.35E-03	N/A	2.79E-03
F11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F12	5.46E-03	1.14E-11	1.14E-11	4.95E-10	6.79E-02	2.50E-05	1.14E-11

Table 5: Friedman test ranking table.

Algorithms	IMRFO	MRFO	HBA	WOA	TLBO	FPA	GWO	PSO
Rank	2.0208	2.2708	2.8333	5.0000	4.4583	7.5417	5.4167	6.4583

experimental parameters are consistent with those above. The test function dimension is 30. If IMRFO is the optimal value, the font is bolded. The experimental results are shown in Table 7.

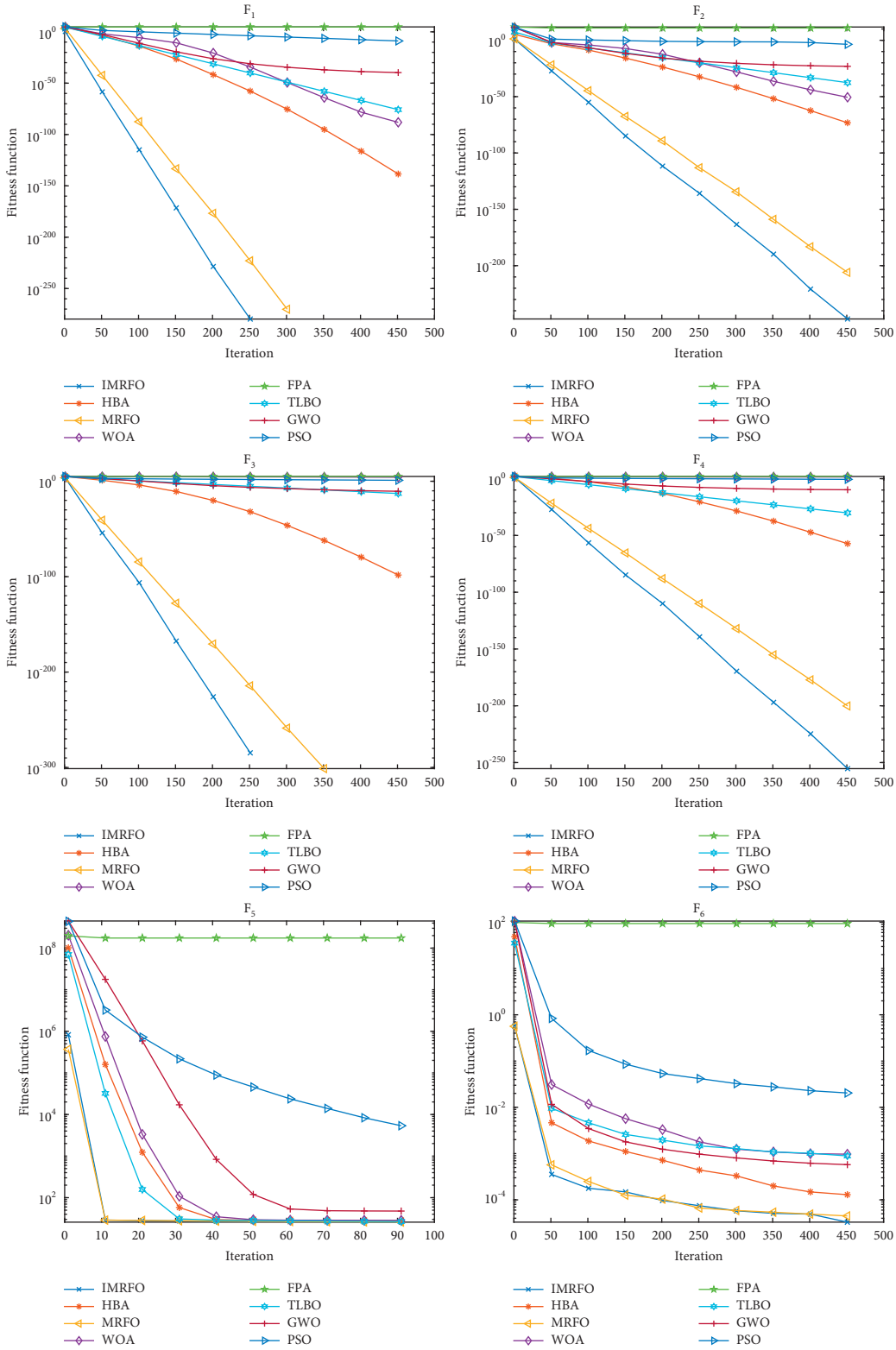
As can be seen from Table 7, IMRFO is the best performer of all variants, and the criteria on each function are the best. IMRFO search accuracy is better than other algorithms and the difference is significant especially in F2, F4, F11. Therefore, it can be seen that the integration of multiple strategies is important, and the validity and feasibility of IMRFO are verified.

5.6. Time Complexity Analysis. Time complexity is an important measure of an algorithm. In order to show an effective improvement, it is necessary to balance the searchability and time complexity of the algorithm. The basic MRFO consists of only three phases, chain feeding, spiral feeding, and empty feeding, where chain feeding and spiral feeding are in the same cycle. Set the population number to N and the maximum number of iterations to T. The dimension is D, so the time complexity of MRFO can be summarized as follows. Macroscopically, the time complexity of swarm intelligence algorithms is the product of

Table 6: Comparison with variants of the algorithm

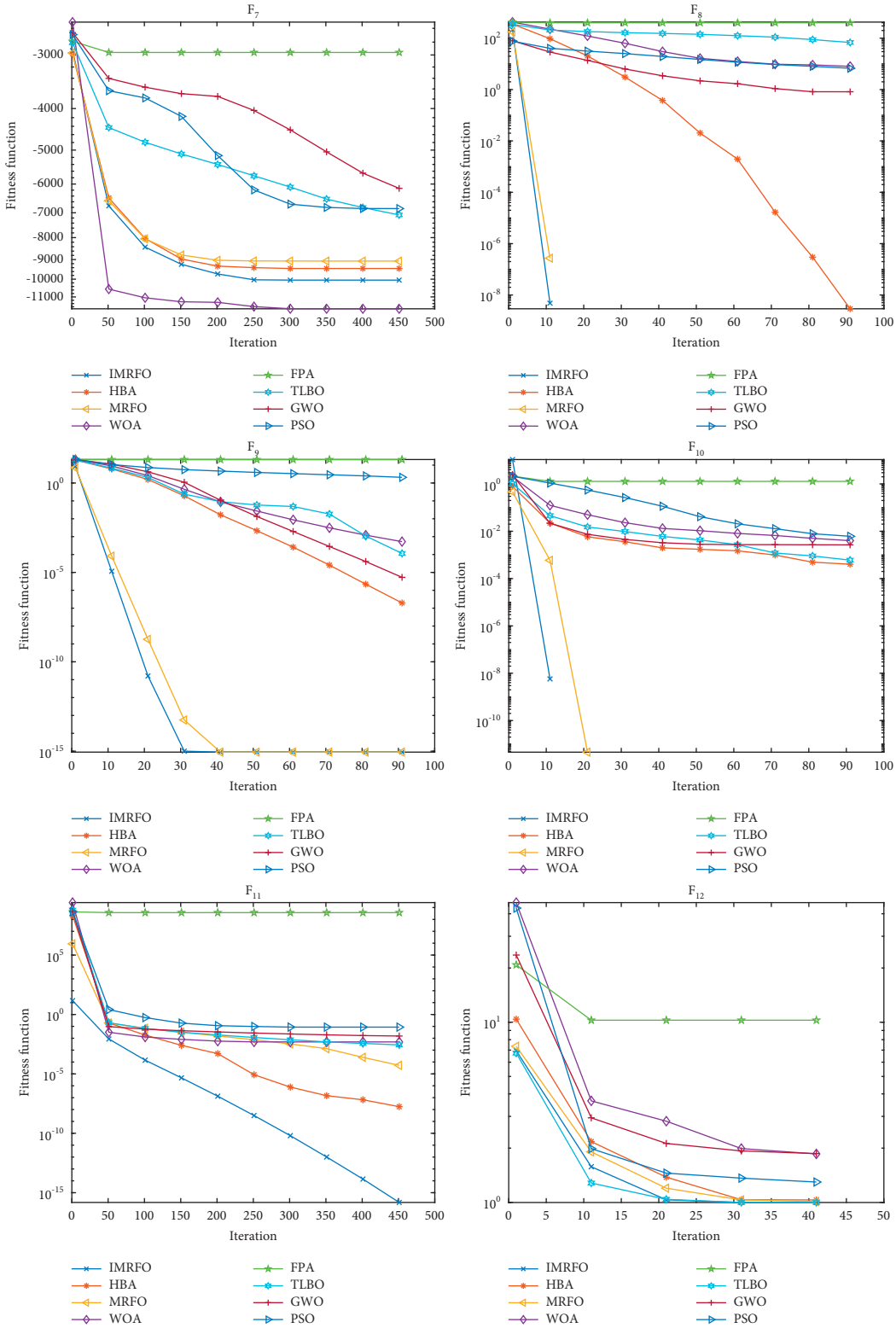
F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	**0**	**0**	**0**
	FACL	$2.6020E+02$	$2.6020E+02$	$4.5704E-13$
	MSSCS	$2.6586E-16$	$1.2320E-14$	$2.2109E-14$
	SACS	$5.8993E-17$	$2.1975E-15$	$2.2919E-15$
	CSsin	$8.1279E-17$	$1.8682E-15$	$2.2000E-15$
$F_2(x)$	IMRFO	**0**	**$4.8161E-277$**	**0**
	FACL	$1.3615E+01$	$1.3615E+01$	$1.7853E-15$
	MSSCS	$2.4907E-09$	$1.0918E-08$	$9.6029E-09$
	SACS	$8.1604E-10$	$3.6463E-09$	$2.3128E-09$
	CSsin	$8.7299E-10$	$3.3430E-09$	$1.7651E-09$
$F_3(X)$	IMRFO	**0**	**0**	**0**
	FACL	$2.3244E+03$	$2.3244E+03$	$2.7422E-12$
	MSSCS	$2.1933E-02$	$6.6588E-01$	$1.2328E+00$
	SACS	$2.0617E-13$	$4.2875E-10$	$7.7897E-10$
	CSsin	$3.4984E-13$	$3.5258E-09$	$1.8741E-08$
$F_4(X)$	IMRFO	**0**	**$3.6613E-283$**	**0**
	FACL	$1.5622E+01$	$1.5622E+01$	$1.4282E-14$
	MSSCS	$9.5018E-05$	$2.8057E-03$	$7.0418E-03$
	SACS	$2.5496E-15$	$2.8163E-12$	$8.9570E-12$
	CSsin	$1.1423E-15$	$1.3641E-12$	$4.9902E-12$
$F_5(X)$	IMRFO	$1.8761E+01$	$1.9089E+01$	$4.2393E-01$
	FACL	$6.2978E+04$	$6.2978E+04$	$7.3126E-11$
	MSSCS	$2.1186E+01$	$2.6991E+01$	$1.5287E+01$
	SACS	$9.5265E-03$	$1.6347E+01$	$1.0434E+01$
	CSsin	$3.1439E+00$	$1.4000E+01$	$4.5249E+00$
$F_6(X)$	IMRFO	**$1.4166E-06$**	**$2.9500E-05$**	**$2.0356E-05$**
	FACL	$4.9638E-01$	$2.6540E+00$	$3.6836E+00$
	MSSCS	$1.0931E-03$	$3.7296E-03$	$1.7917E-03$
	SACS	$2.7015E-04$	$1.2413E-03$	$6.5230E-04$
	CSsin	$2.5773E-04$	$1.2047E-03$	$6.2553E-04$
$F_7(X)$	IMRFO	$-1.0536E+04$	$-1.0050E+04$	$4.2525E+02$
	FACL	$-3.2588E+03$	$-3.2588E+03$	$3.6563E-12$
	MSSCS	$-1.2409E+04$	$-1.2023E+04$	$1.7642E+02$
	SACS	$-1.2337E+04$	$-1.2040E+04$	$1.6571E+02$
	CSsin	$-1.2197E+04$	$-1.1773E+04$	$2.2010E+02$
$F_8(X)$	IMRFO	**0**	**0**	**0**
	FACL	$1.3469E+02$	$1.3469E+02$	$1.9995E-13$
	MSSCS	$1.3682E+02$	$1.3682E+02$	$2.2852E-13$
	SACS	$1.3558E+01$	$1.9318E+01$	$3.2639E+00$
	CSsin	$1.3082E+01$	$1.7173E+01$	$2.4068E+00$
$F_9(X)$	IMRFO	**$8.8818E-16$**	**$8.8818E-16$**	**0**
	FACL	$9.5803E+00$	$9.5803E+00$	$1.4282E-14$
	MSSCS	$7.8366E-09$	$3.3808E-08$	$2.3225E-08$
	SACS	$6.6614E-09$	$3.5846E-08$	$2.6071E-08$
	CSsin	$9.4992E-09$	$4.9013E-08$	$3.5110E-08$
$F_{10}(X)$	IMRFO	**0**	**0**	**0**
	FACL	$1.6979E+01$	$1.6979E+01$	$2.4994E-14$
	MSSCS	$6.6613E-16$	$8.2473E-04$	$4.4988E-03$
	SACS	$1.6653E-15$	$3.2272E-03$	$9.0416E-03$
	CSsin	$6.6613E-16$	$1.8055E-03$	$8.5946E-03$
$F_{11}(X)$	IMRFO	**$7.3400E-20$**	**$1.2392E-18$**	**$1.2894E-18$**
	FACL	$8.7100E+00$	$8.7100E+00$	$1.4282E-14$
	MSSCS	$1.5139E-07$	$5.4848E-07$	$3.2760E-07$
	SACS	$2.2637E-07$	$4.8725E-07$	$3.3696E-07$
	CSsin	$1.7143E-07$	$5.0219E-07$	$3.9414E-07$
$F_{12}(X)$	IMRFO	**$9.9800E-01$**	**$9.9800E-01$**	**0**
	FACL	$1.9920E+00$	$1.9920E+00$	$4.4633E-15$
	MSSCS	—	—	—
	SACS	$9.9800E-01$	$9.9800E-01$	0
	CSsin	$9.9800E-01$	$9.9800E-01$	0

The optimal value is shown in bold.



(a)

Figure 4: Continued.



(b)

Figure 4: Average convergence of each algorithm.

Table 7: Test table for each combination algorithm.

F	Algorithm	Best	Mean	Std
$F_1(x)$	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0
$F_2(x)$	IMRFO	**0**	**4.8161E-277**	**0**
	MRFO-I	2.2366E-239	2.3660E-230	0
	MRFO-II	1.4788E-239	3.6180E-231	0
	MRFO-III	6.7519E-240	2.1114E-231	0
	MRFO-IV	4.9710E-237	1.7202E-229	0
	MRFO-V	3.4101E-236	1.0691E-229	0
	MRFO-VI	6.6058E-239	8.1070E-230	0
$F_3(X)$	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0
$F_4(X)$	IMRFO	**0**	**3.6613E-283**	**0**
	MRFO-I	7.1957E-231	9.1743E-224	0
	MRFO-II	5.5959E-232	2.5385E-224	0
	MRFO-III	2.2845E-232	2.3915E-224	0
	MRFO-IV	1.7079E-231	6.1960E-224	0
	MRFO-V	3.1285E-233	3.7719E-225	2.0880
	MRFO-VI	3.3266E-233	2.8006E-224	8.6554E-02
$F_5(X)$	IMRFO	**1.8761 E+01**	**1.9089 E+01**	4.2393E-01
	MRFO-I	1.9160 E+01	2.0228 E+01	4.7486E-01
	MRFO-II	1.9205 E+01	1.9904 E+01	4.1531E-01
	MRFO-III	1.9050 E+01	2.0040 E+01	4.5484E-01
	MRFO-IV	1.9054 E+01	1.9923 E+01	4.8479E-01
	MRFO-V	1.8959 E+01	2.0027 E+01	5.1658E-01
	MRFO-VI	1.8883 E+01	1.9994 E+01	6.7667E-01
$F_6(X)$	IMRFO	**1.4166E-06**	**2.9500E-05**	**2.0356E-05**
	MRFO-I	3.3254E-06	4.0901E-05	2.9878E-05
	MRFO-II	1.7086E-06	4.2369E-05	3.8592E-05
	MRFO-III	2.9884E-06	4.3133E-05	4.4862E-05
	MRFO-IV	5.3342E-06	5.1792E-05	5.3359E-05
	MRFO-V	8.3930E-06	6.0626E-05	4.2128E-05
	MRFO-VI	4.8785E-06	5.3014E-05	4.7227E-05
$F_7(X)$	IMRFO	**-1.0536 E+04**	**-1.0050 E+04**	**4.2525 E+02**
	MRFO-I	-9.7467 E+03	-8.7240 E+03	6.4405 E+02
	MRFO-II	-9.9243 E+03	-8.7951 E+03	6.3557 E+02
	MRFO-III	-9.8848 E+03	-8.8964 E+03	5.5148 E+02
	MRFO-IV	-1.0102 E+04	-8.7470 E+03	6.3365 E+02
	MRFO-V	-1.0359 E+04	-8.9240 E+03	7.0323 E+02
	MRFO-VI	-1.0082 E+04	-9.0438 E+03	5.6962 E+02
$F_8(X)$	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0

Table 7: Continued.

F	Algorithm	Best	Mean	Std
$F_9(X)$	IMRFO	**8.8818E-16**	**8.8818E-16**	**0**
	MRFO-I	8.8818E-16	8.8818E-16	0
	MRFO-II	8.8818E-16	8.8818E-16	0
	MRFO-III	8.8818E-16	8.8818E-16	0
	MRFO-IV	8.8818E-16	8.8818E-16	0
	MRFO-V	8.8818E-16	8.8818E-16	0
	MRFO-VI	8.8818E-16	8.8818E-16	0
$F_{10}(X)$	IMRFO	**0**	**0**	**0**
	MRFO-I	0	0	0
	MRFO-II	0	0	0
	MRFO-III	0	0	0
	MRFO-IV	0	0	0
	MRFO-V	0	0	0
	MRFO-VI	0	0	0
$F_{11}(X)$	IMRFO	**7.3400E-20**	1.2392E-18	1.2894E-18
	MRFO-I	1.9606E-19	1.5295E-18	1.7575E-18
	MRFO-II	1.8111E-19	1.4964E-18	1.1884E-18
	MRFO-III	1.2951E-19	2.0136E-18	2.1218E-18
	MRFO-IV	3.8758E-19	2.7853E-18	4.2987E-18
	MRFO-V	1.1870E-19	1.1469E-18	1.0703E-18
	MRFO-VI	9.2691E-20	1.8700E-18	2.1327E-18
$F_{12}(X)$	IMRFO	**9.9800E-01**	**9.9800E-01**	**0**
	MRFO-I	9.9800E-01	9.9800E-01	0
	MRFO-II	9.9800E-01	9.9800E-01	0
	MRFO-III	9.9800E-01	9.9800E-01	0
	MRFO-IV	9.9800E-01	9.9800E-01	0
	MRFO-V	9.9800E-01	9.9800E-01	0
	MRFO-VI	9.9800E-01	9.9800E-01	0

Bold font is to clearly see the advantages of the algorithm.

population number, iteration number, and dimension. Therefore, the time complexity of IMRFO is O (TND), just like other algorithms.

Microscopically, MRFO can be calculated as follows:

$$\begin{aligned} O(\text{IMRFO}) &= O(T(O(\text{cyclone foraging} + \text{chain foraging}) \\ &\quad + O(\text{somersault foraging}))) \\ &= O(T(ND + ND)) = O(TND). \end{aligned} \tag{11}$$

Set the calculation time of introducing RWL to be t1, the calculation time of introducing Lévy flight to be t_2, the calculation time of using two learning factors to be t_3, and the other calculations are ignored.

IMRFO can be summarized as follows:

$$\begin{aligned} O(\text{MRFO}) &= O\big(T\big(O(\text{cyclone foraging} + \text{chain foraging}) + O(t1) + O(\text{somersault foraging}) + O(t_2) + O(t_3)\big)\big) \\ &= O(TND). \end{aligned} \tag{12}$$

Therefore, it can be seen that the time complexity of IMRFO has not changed fundamentally. A small increase in the number of iterations can be ignored. These increases will be of great significance if the optimization capability of the algorithm is effectively improved.

6. Image Segmentation Experiments

At present, image processing has been applied in many fields, and the image on land has been well developed, but it still has research value in underwater images. So in order to show the research value, eight underwater images are selected as test images. From the literature [54], select PSO, DPSO, sparrow search algorithm (SSA) [55], Modified sparrow search algorithm (MSSA) [56], ABC, MRFO, WOA, TLBO, FPA, IMRFO 9 algorithms optimize K-means algorithm and traditional K-means algorithm for image segmentation. MSSA is a newly proposed K-means based algorithm, and other algorithms have been successfully applied to image segmentation problems in recent years. Because the K-means clustering algorithm has a strong dependence on K values, improper selection of K values will have a great impact on the results, and K-means clustering algorithm has a strong dependence

on K values. Set the value of k to 3 to avoid interference from unrelated factors. The general parameters of the algorithm are population size of 30 and the maximum number of iterations of 100. Each algorithm divides the image as shown in Figure 5 and Figure 6. The first line in Figure 5 and Figure 6 represents the original image, and each subsequent line represents the segmentation effect of each algorithm.

It is impossible to see the difference between each algorithm in image segmentation by human eyes. Therefore, three commonly used image segmentation metrics, PSNR, SSIM, and FSIM, are selected to measure the quality of each algorithm.

Peak Signal-to-Noise Ratio (PSNR) is mainly used to measure the difference between the segmented image and the original image. The formula is as follows [57]:

$$\begin{aligned} \mathrm{PSNR} &= 20.\log_{10}\left(\frac{255}{\mathrm{RMSE}}\right), \\ \mathrm{RMSE} &= \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{Q}\left(I(i,j)-Seg(i,j)\right)^2}{M\times Q}}. \end{aligned} \tag{13}$$

In formula (12) and (13), RMSE represents the root mean square error of the pixels; $M\times Q$ represents the size of the image; $I(i, j)$ represents the pixel gray value of the original image; $\mathrm{Seg}(i, j)$ represents the pixel gray value of the segmented image. The larger the PSNR value, the better the segmented image quality. Generally speaking, PSNR higher than 40 dB indicates excellent image quality (indicating that it is very close to the original image). At 30–40db, it usually indicates that the image quality is good (indicating that the distortion is perceptible but acceptable).

Structural Similarity (SSIM) is used to measure the similarity between the original image and the segmented image. The larger the SSIM, the better the segmented result. SSIM is defined as follows [58]:

$$\mathrm{SSIM} = \frac{\left(2\mu_I\mu_{\mathrm{seg}} + c_1\right)\left(2\sigma_{I,\mathrm{seg}} + c_2\right)}{\left(\mu_I^2 + \mu_{\mathrm{seg}}^2 + c_1\right)\left(\sigma_I^2 + \sigma_{\mathrm{seg}}^2 + c_2\right)}. \tag{14}$$

In formula (14), μ_I and μ_{seg} represent the average value of the original image and the segmented image; σ_I and σ_{seg} represent the standard deviation of the original image and the segmented image, respectively.; $\sigma_{I,\mathrm{seg}}$ represents the covariance between the original image and the segmented image; c_1, c_2 are constantly used to ensure stability. SSIM value range [0,1]. The larger the value, the smaller the image distortion.

Feature similarity index mersure (FSIM) is a measure of the characteristic similarity between the original image and the quality of the segmentation, used to evaluate local structure and provide contrast information. The value range of FSIM is [0,1], and the closer the value is to 1, the better the segmentation effect. FSIM is defined as follows [59]:

$$\begin{aligned} \mathrm{FSIM} &= \frac{\sum_{l\in\Omega} S_L(X)PC_m(X)}{\sum_{l\in\Omega} PC_m(X)}, \\ SL(X) &= S_{PC}(X)S_G(X), \\ S_{PC}(X) &= \frac{2PC_1(X)PC_2(X) + T_1}{PC_1^2(X)PC_2^2(X) + T_1}, \\ PC(X) &= \frac{E(X)}{\left(\varepsilon + \sum_m A_n(X)\right)}. \\ G &= \sqrt{G_x^2 + G_y^2}, \\ S_G(X) &= \frac{2G_1(X)G_2(X) + T_2}{G_1^2(X)G_2^2(X) + T_2}, \end{aligned} \tag{15}$$

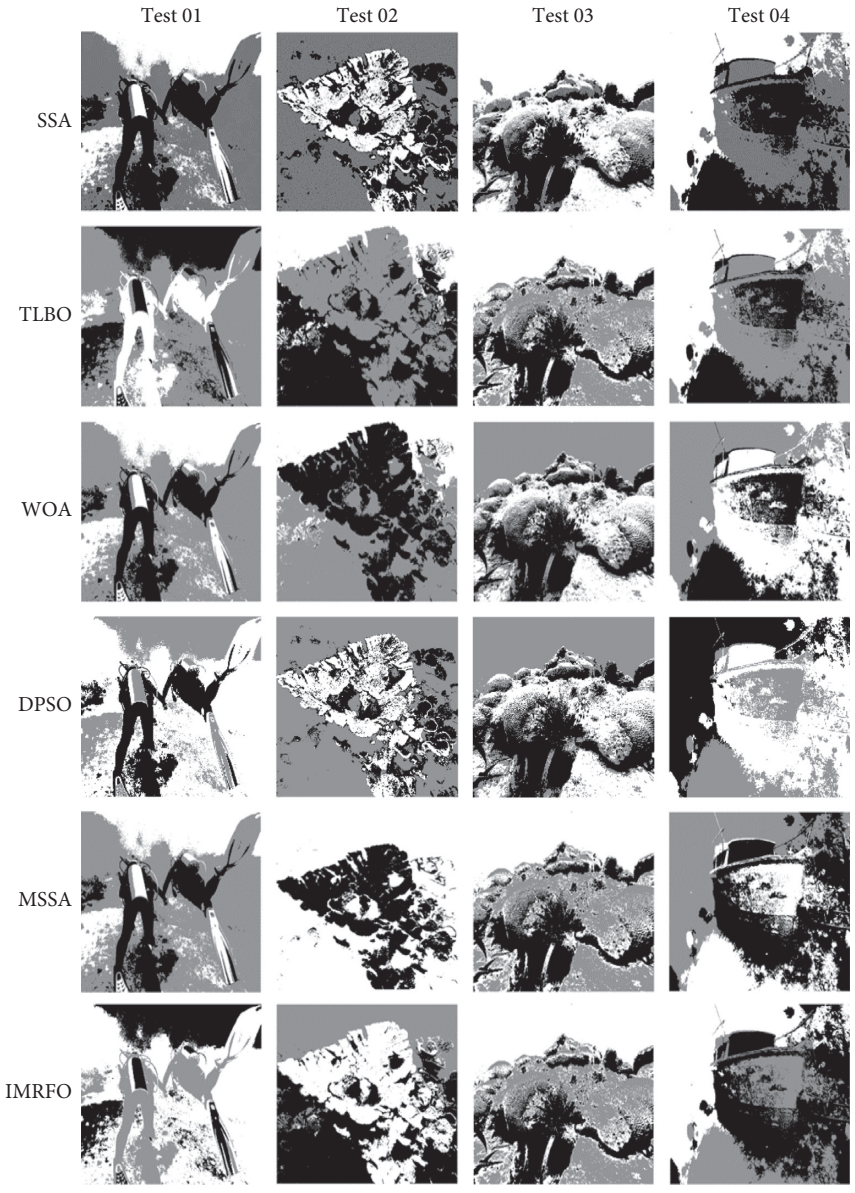
In the above formula, Ω is all the pixel regions of the original image; $S_L(X)$ is the similarity score; $PC_m(X)$ is the phase consistency measure; T_1 and T_2 are constants; G is the gradient descent; $E(x)$ is the response vector size at position X and the scale is n; ε is a very small value; $An(X)$ is the local size at scale n.

Run each algorithm 10 times, and the average and average running time of the partitioned metrics are shown in Table 8.

Simply from the naked eye, the image after IMRFO segmentation in Figures 5 and Figure 6 is clearer. Some algorithms have a rough segmentation effect and have appeared a blurry phenomenon. From Table 7, it can be seen that the segmentation index of IMRFO has a greater advantage, especially in test01 and test03-08, where more than two indexes are optimal. For example, the FSIM index in test07 reaches 0.97, SSIM in test08 reaches 0.87, which has a significant advantage over other algorithms. When the performance indicator is not optimal, IMRFO is still close to the optimal value. For example, in test01, the SSIM index of WOA is 0.7488, while that of IMRFO is 0.7479, which is close to the optimal value. In test06, the PSNR of ABC was 43.3715, and that of IMRFO was 43.1626. Therefore, both subjective visual effect and measurement result of IMRFO is better than other algorithms, which can prove a good segmentation effect. It also indirectly proves the good search performance of IMRFO, solves the problem that MRFO is easy to fall into local optimal solution and K-means has the disadvantage of being sensitive to the initial clustering center, which results in an excellent initial clustering center and further improves the image segmentation quality. On the other hand, the running time of K-means segmentation is the least, but the quality is the worst. The operation of other algorithms is large and the effect is obvious. The IMRFO does have a time disadvantage, to be expected, as it takes more time to accurately scan the solution in space.

(a)

Figure 5: Continued.



(b)

Figure 5: Segmentation effect of test 01–4.

7. Summary of Results

MRFO relies on group behavior to find food, so it lacks flexibility and is prone to fall into local optimum. In the existing work, most scholars can not solve such problems well. In order to improve the searching ability of MRFO, an improved algorithm for bats is presented, which uses Lévy flight, random walk learning, and learning factors.

The current experimental work is summarized as follows:

(1) Comparing IMRFO with some basic algorithms on 12 standard test functions shows that the algorithm has certain advantages.

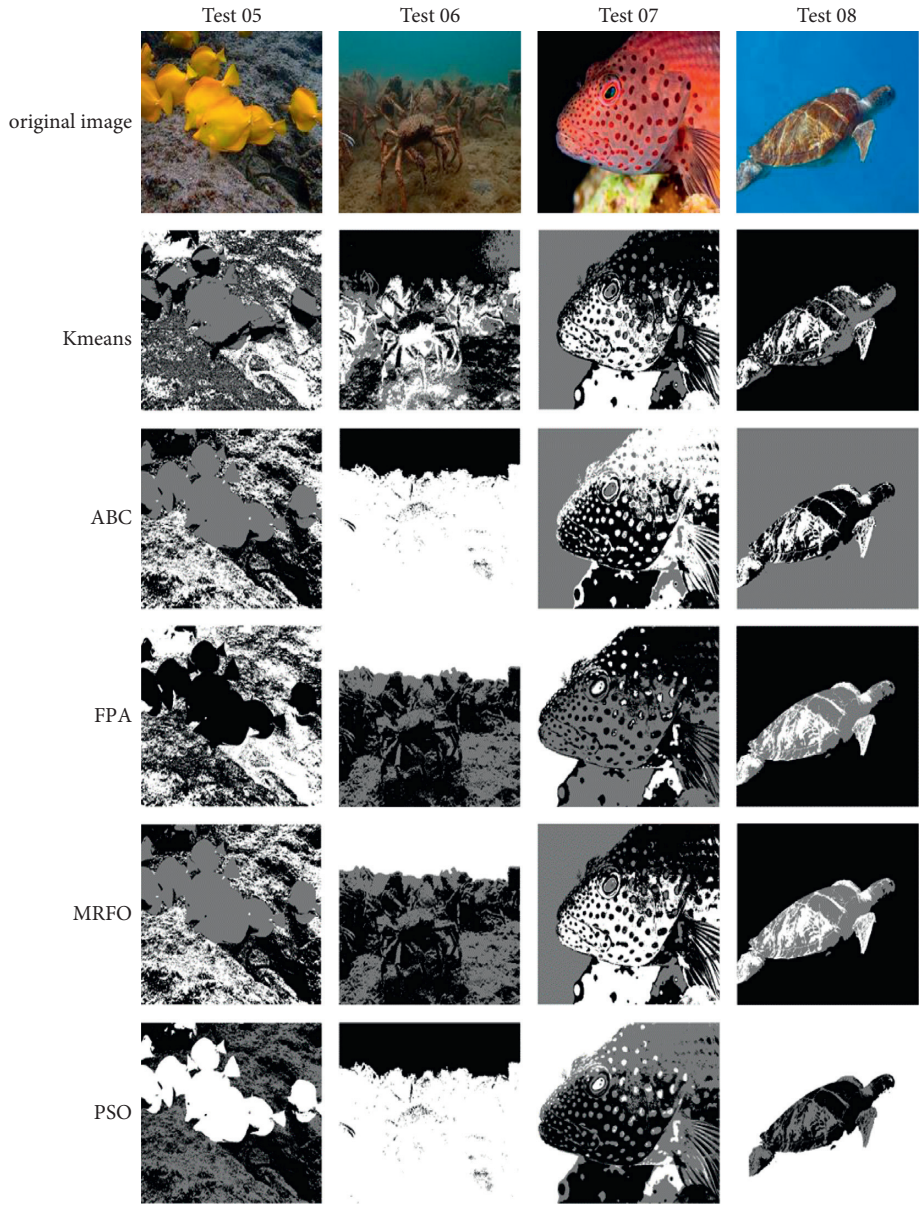
(2) Two statistical tests are used to verify the universality of IMRFO and show a good search ability.

(3) The convergence of each algorithm in each function is given, and the result shows that IMRFO has a good convergence rate.

(4) To further verify the performance of the algorithm, IMRFO is compared with the recently proposed variants of the algorithm, and the results show that IMRFO has an obvious advantage in most functions.

(5) In order to verify the validity and value of the three combinations of strategies, ablation experiments were carried out. The results show that IMRFO is better than other combinations of strategies, highlighting the practical value of IMRFO.

(6) Eight underwater images were used to verify the effect of IMRFO optimized K-means image



(a)

Figure 6: Continued.

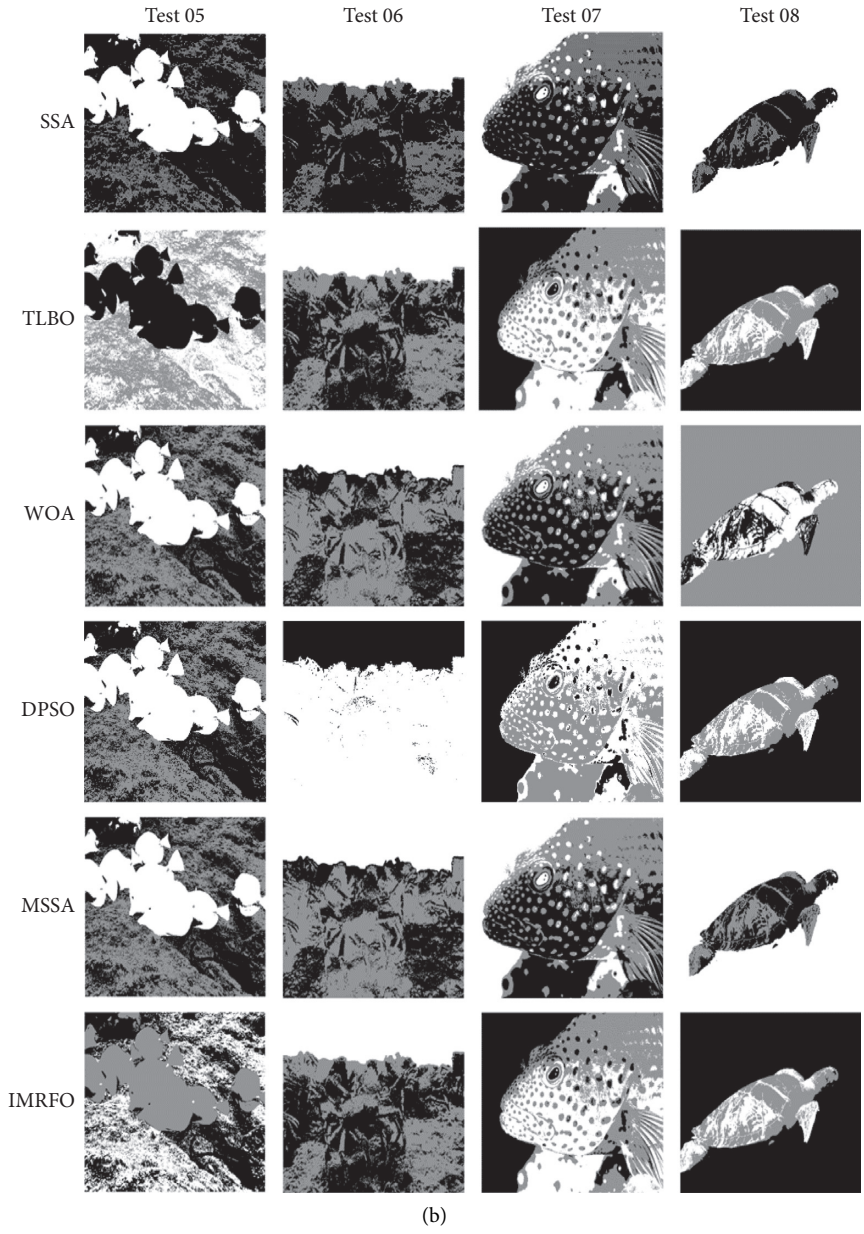


Figure 6: Segmentation effect of test 05–8.

Table 8: Segmentation effect tables for each algorithm.

Image	Index	K-means	PSO	MRFO	DPSO	MSSA	IMRFO
Test 01	PSNR	8.0602	42.9051	42.9934	42.8040	43.4789	**44.4989**
	SSIM	0.0525	0.7354	0.7412	0.7370	0.7671	0.7479
	FSIM	0.5556	0.9478	0.9476	0.9571	0.9467	**0.9579**
	Ave time	**2.5886**	39.2606	76.7789	38.4986	50.6846	76.3475
Test 02	PSNR	6.4532	44.0008	43.3155	44.0595	43.9037	43.7004
	SSIM	0.0306	0.7975	0.7788	**0.8027**	0.7877	0.7970
	FSIM	0.4934	0.9464	0.9473	0.9410	0.9508	**0.9601**
	Ave time	**5.9745**	43.9744	78.42064	38.7315	50.7706	74.5214
Test 03	PSNR	6.6314	43.1602	43.3925	43.3760	43.2782	**43.7128**
	SSIM	0.0273	0.7615	0.7770	0.7655	0.7678	**0.7894**
	FSIM	0.4249	0.9400	0.9470	0.9424	**0.9506**	0.9430
	Ave time	**2.4941**	42.6951	75.7196	38.5346	50.7550	74.7570

Table 8: Continued.

Image	Index	K-means	PSO	MRFO	DPSO	MSSA	IMRFO
Test 04	PSNR	7.6085	43.0707	42.7419	43.2700	42.6999	43.1534
	SSIM	0.0461	0.7559	0.7396	0.7651	0.7380	**0.7808**
	FSIM	0.4920	0.9351	0.9435	0.9334	0.9397	**0.9466**
	Ave time	**5.8564**	41.2824	77.9430	38.4732	52.1376	75.2728
Test 05	PSNR	7.7941	43.1295	42.9913	43.2999	42.6999	**43.5453**
	SSIM	0.0187	0.7542	0.7503	0.7637	0.7380	**0.7726**
	FSIM	0.4011	0.9551	0.9566	0.9596	0.9397	**0.9634**
	Ave time	**2.2253**	37.4426	73.5829	37.6144	48.2604	73.7482
Test 06	PSNR	9.6892	42.9705	42.7052	43.539	42.8316	43.1626
	SSIM	0.0393	0.7375	0.7309	0.7385	0.7338	**0.7517**
	FSIM	0.5301	0.9501	0.9567	0.9480	0.9474	**0.9604**
	Ave time	**1.2299**	37.5475	72.6558	39.0477	48.7618	74.5432
Test 07	PSNR	7.82	42.7995	42.8791	72.8223	42.7924	**43.0107**
	SSIM	0.1071	0.7357	0.7399	0.7377	0.7414	0.7409
	FSIM	0.4460	0.9625	0.9584	0.9667	0.9558	**0.9734**
	Ave time	**2.3464**	39.1573	74.3283	38.1076	49.2879	74.5704
Test 08	PSNR	8.2165	44.1114	43.5191	44.5560	42.8239	**46.4141**
	SSIM	0.0210	0.7681	0.7417	0.7835	0.7185	**0.8710**
	FSIM	0.6614	0.8259	0.8130	0.8064	0.8444	0.8340
	Ave time	**1.2530**	38.8315	74.7158	38.2203	48.7720	75.4689
F	*index*	*WOA*	*ABC*	*TLBO*	*FPA*	*SSA*	*IMRFO*
Test 01	PSNR	43.1176	43.1018	42.8136	43.0370	43.4300	**44.4989**
	SSIM	**0.7488**	0.7473	0.7341	0.7407	0.7545	0.7479
	FSIM	0.9476	0.9441	0.9438	0.9441	0.9466	**0.9579**
	ave time	38.3036	39.4908	74.8217	40.1402	39.1653	76.3475
Test 02	PSNR	**44.1919**	43.5238	43.4408	42.9663	43.4401	43.7004
	SSIM	**0.8006**	0.7836	0.7852	0.7622	0.7808	0.7970
	FSIM	0.9482	0.9585	0.9557	0.9487	0.9531	**0.9601**
	ave time	38.8852	40.4172	75.3121	39.2922	42.0149	74.5214
Test 03	PSNR	43.4085	43.4408	43.0703	43.2635	43.3470	**43.7128**
	SSIM	0.7746	0.7790	0.7641	0.7731	0.7738	**0.7894**
	FSIM	0.9422	0.9453	0.9452	0.9442	0.9503	0.9430
	ave time	38.7937	42.8378	75.0803	41.8402	40.2512	74.7570
Test 04	PSNR	**43.3877**	43.2727	43.2877	42.7467	42.4343	43.1534
	SSIM	0.7600	0.7622	0.7630	0.7410	0.7256	**0.7808**
	FSIM	0.9426	0.9402	0.9403	0.9400	0.9380	**0.9466**
	ave time	39.2289	40.4403	74.4058	39.5984	39.0673	75.2728
Test 05	PSNR	42.9238	42.9262	42.9594	42.8405	42.9593	**43.5453**
	SSIM	0.7518	0.7514	0.7497	0.7480	0.7497	**0.7726**
	FSIM	0.9607	0.9605	0.9568	0.9586	0.9568	**0.9634**
	ave time	39.0783	38.9743	74.9810	38.6717	40.5572	73.7482
Test 06	PSNR	42.9003	**43.3715**	42.5365	42.5136	43.1584	43.1626
	SSIM	0.7356	0.7501	0.7107	0.7147	10.7503	**0.7517**
	FSIM	0.9396	0.9503	0.9444	0.9442	0.9511	**0.9604**
	ave time	37.5257	38.4848	76.3897	38.1784	37.8186	74.5432
Test 07	PSNR	42.8455	42.8353	42.8986	42.8399	42.8093	**43.0107**
	SSIM	0.7410	0.7386	0.7409	**0.7450**	0.7397	0.7409
	FSIM	0.9426	0.9551	0.9566	0.9367	0.9491	**0.9734**
	ave time	38.7532	39.6091	74.4956	38.6593	38.118674	74.5704
Test 08	PSNR	44.0623	45.5442	43.2560	45.3984	43.7115	**46.4141**
	SSIM	0.7663	0.8221	0.7315	0.8182	0.7545	**0.8710**
	FSIM	0.8381	0.8398	0.8196	0.8263	0.8316	0.8340
	ave time	39.3148	39.1180	76.6363	39.1871	38.8752	75.4689

The bold font here is to see the advantages and disadvantages of IMRFO. Although IMRFO has good performance, it has no advantage in time.

segmentation. The results show that the IMRFO optimized image segmentation quality is good and has a rational segmentation index in multiple images.

In summary, several experiments have demonstrated that IMRFO is a challenging new variant of the algorithm. IMRFO has shown good results in several test functions and

image segmentation, but the optimization results in some functions and images need to be improved. More work is waiting to improve its optimization capability.

8. Conclusion and Future Works

In order to improve the shortcomings of K-means image segmentation and its vulnerability to local optimization, this paper presents a K-means image segmentation method based on IMRFO. IMRFO uses Lévy flight to improve the individual searchability, proposes random walk learning to prevent the premature phenomenon of the algorithm, and finally uses learning factor to improve the convergence accuracy of the algorithm so as to improve the search of the algorithm. The validity and feasibility of IMRFO are verified by 12 test functions, and through 8 underwater image data sets, it can be seen that IMRFO has a good segmentation effect and is superior to other algorithms proposed in recent years under several indicators.

Although IMRFO has good segmentation advantages in eight images, it does not achieve the best three criteria for all images. From the experimental point of view, IMRFO is only the best in test 05, while the other test pictures are basically the two best. On the other hand, the running time of each algorithm is too large, and the accuracy is at the expense of time. In the future, we will improve the image quality from the following three aspects.

(1) Comprehensively improve the three performance indicators, making the three indicators the best

(2) Balance the time and the search ability of the algorithm to get the best performance in an effective time

(3) It can be used in agricultural, aerospace, medical, and other scenarios so that the algorithm can play a suitable role in different environments

Data Availability

Some data of this study are confidential, so the experimental data cannot be uploaded. These data can be obtained from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China under Grant 61672121.

References

[1] E. Ardizzone, R. Pirrone, and O. Gambino, "Fuzzy c-means segmentation on brain mr slices corrupted by rf-inhomogeneity," in *Proceedings of the International Workshop on Fuzzy Logic and Applications*, Springer, Berlin, Heidelberg, 2007.

[2] W. Li, G.-G. Wang, and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3781–3799, 2021.

[3] R. C. Hrosik, E. Tuba, E. Dolicanin, R. Jovanovic, and M. Tuba, "Brain image segmentation based on firefly algorithm combined with k-means clustering," *Studies in Informatics and Control*, vol. 28, pp. 167–176, 2019.

[4] H. Li, H. He, and Y. Wen, "Dynamic particle swarm optimization and K-means clustering algorithm for image segmentation," *Optik*, vol. 126, no. 24, pp. 4817–4822, 2015.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[6] S. Kapoor, I. Zeya, C. Singhal, and S. J. Nanda, "A grey wolf optimizer based automatic clustering algorithm for satellite image segmentation," *Procedia Computer Science*, vol. 115, 2017.

[7] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[8] G.-G Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, IEEE, Bali, Indonesia, 2015.

[9] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.

[10] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[11] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, vol. 5, IEEE, Orlando, FL, USA, October 1997.

[12] X. Xia, L. Gui, F. Yu et al., "Triple archives particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 50, pp. 4862–4875, 2019.

[13] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, pp. 65–85, 1994.

[14] S.-C. Liu, Z.-G. Chen, Z.-H. Zhan, S.-W. Jeon, S. Kwong, and J. Zhang, "Many-objective job-shop scheduling: a multiple populations for multiple objectives-based genetic algorithm approach," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–15, 2021.

[15] S. Das and N. S. Ponnuthurai, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 4–31, 2010.

[16] Z. H. Zhan, Z. J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–15, 2019.

[17] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing*, IEEE, Coimbatore, India, 2010.

[18] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[19] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[20] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications," *Engineering Applications of Artificial Intelligence*, vol. 87, Article ID 103300, 2020.

[21] A. Fathy, H. Rezk, and D. Yousri, "A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm," *Solar Energy*, vol. 207, pp. 305–316, 2020.

[22] E. H. Houssein, G. N. Zaki, A. A. Z. Diab, and E. M. Younis, "An efficient Manta Ray Foraging Optimization algorithm for parameter extraction of three-diode photovoltaic model," *Computers & Electrical Engineering*, vol. 94, Article ID 107304, 2021.

[23] E. H. Houssein, I. E. Ibrahim, N. Neggaz, M. Hassaballah, and Y. M. Wazery, "An efficient ECG arrhythmia classification method based on Manta ray foraging optimization," *Expert Systems with Applications*, vol. 181, Article ID 115131, 2021.

[24] M. Aly and H. Rezk, "A MPPT based on optimized FLC using manta ray foraging optimization algorithm for thermo-electric generation systems," *International Journal of Energy Research*, vol. 45, no. 9, pp. 13897–13910, 2021.

[25] M. G. Hemeida, A. A. Ibrahim, A.-A. A. Mohamed, S. Alkhalaf, and A. M. B. El-Dine, "Optimal allocation of distributed generators DG based Manta Ray Foraging Optimization algorithm (MRFO)," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 609–619, 2021.

[26] M. G. Hemeida, S. Alkhalaf, A.-A. A. Mohamed, A. A. Ibrahim, and T. Senjyu, "Distributed generators optimization based on multi-objective functions using manta rays foraging optimization algorithm (MRFO)," *Energies*, vol. 13, no. 15, p. 3847, 2020.

[27] A. M. Shaheen, A. R. Ginidi, R. A. El-Sehiemy, and S. S. M. Ghoneim, "Economic power and heat dispatch in cogeneration energy systems using manta ray foraging optimizer," *IEEE Access*, vol. 8, pp. 208281–208295, 2020.

[28] U. C. Ben, A. E. Akpan, C. C. Mbonu, and E. D. Ebong, "Novel methodology for interpretation of magnetic anomalies due to two-dimensional dipping dikes using the Manta Ray Foraging Optimization," *Journal of Applied Geophysics*, vol. 192, Article ID 104405, 2021.

[29] O. Akdag and C. Yeroglu, "Optimal directional overcurrent relay coordination using MRFO algorithm: a case study of adaptive protection of the distribution network of the Hatay province of Turkey," *Electric Power Systems Research*, vol. 192, Article ID 106998, 2021.

[30] K. K. Ghosh, R. Guha, S. K. Bera, N. Kumar, and R. Sarkar, "S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem," *Neural Computing & Applications*, vol. 33, no. 17, pp. 11027–11041, 2021.

[31] F. A Alturki, H. Mh Farh, A. A Al-Shamma'a, and K. AlSharabi, "Techno-economic optimization of small-scale hybrid energy systems using manta ray foraging optimizer," *Electronics*, vol. 9, p. 12, 2020.

[32] K. Elmaadawy, M. Abd Elaziz, A. H. Elsheikh, A. Moawad, B. Liu, and S. Lu, "Utilization of random vector functional link integrated with manta ray foraging optimization for effluent prediction of wastewater treatment plant," *Journal of Environmental Management*, vol. 298, Article ID 113520, 2021.

[33] M. Abd Elaziz, D. Yousri, M. A. Al-qaness, A. M. AbdelAty, A. G. Radwan, and A. A. Ewees, "A Grunwald–Letnikov based Manta ray foraging optimizer for global optimization and image segmentation," *Engineering Applications of Artificial Intelligence*, vol. 98, Article ID 104105, 2021.

[34] M. H. Hassan, E. H. Houssein, M. A. Mahdy, and S. Kamel, "An improved manta ray foraging optimizer for cost-effective emission dispatch problems," *Engineering Applications of Artificial Intelligence*, vol. 100, Article ID 104155, 2021.

[35] H. Xu, H. Song, C. Xu, X. Wu, and N. Yousefi, "Exergy analysis and optimization of a HT-PEMFC using developed manta ray foraging optimization algorithm," *International Journal of Hydrogen Energy*, vol. 45, no. 55, pp. 30932–30941, 2020.

[36] E. H. Houssein, M. M. Emam, and A. A. Ali, "Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images," *Neural Computing & Applications*, vol. 33, no. 24, pp. 16899–16919, 2021.

[37] B. Jena, M. K. Naik, R. Panda, and A. Abraham, "Maximum 3D Tsallis entropy based multilevel thresholding of brain MR image using attacking Manta Ray foraging optimization," *Engineering Applications of Artificial Intelligence*, vol. 103, Article ID 104293, 2021.

[38] M. Micev, M. Ćalasan, Z. M. Ali, H. M. Hasanien, and S. H. E. Abdel Aleem, "Optimal design of automatic voltage regulation controller using hybrid simulated annealing - manta ray foraging optimization algorithm," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 641–657, 2021.

[39] S. Ekinci, D. Izci, and B. Hekimoğlu, "Optimal FOPID speed control of DC motor via opposition-based hybrid manta ray foraging optimization and simulated annealing algorithm," *Arabian Journal for Science and Engineering*, vol. 46, no. 2, pp. 1395–1409, 2021.

[40] H. . Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.

[41] R. Jensi and G. W. Jiji, "An enhanced particle swarm optimization with levy flight for global optimization," *Applied Soft Computing*, vol. 43, pp. 248–261, 2016.

[42] Y. Liu and B. Cao, "A novel ant colony optimization algorithm with Levy flight," *IEEE Access*, vol. 8, pp. 67205–67213, 2020.

[43] X. Li, A. Engelbrecht, and M. G. Epitropakis, *Benchmark functions for CEC'2013 special session and competition on niching methods for multimodal function optimization*, RMIT University, Evolutionary Computation and Machine Learning Group, Australia, 2013.

[44] M. Molga and C. Smutnicki, "Test functions for optimization needs," *Test Functions for Optimization Needs*, vol. 101, p. 48, 2005.

[45] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey Badger Algorithm: new metaheuristic algorithm for solving optimization problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84–110, 2022.

[46] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[47] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[48] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proceedings of the International Conference on Unconventional Computing and Natural Computation*, 2012.

[49] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[50] H. Peng, Z. Zeng, C. Deng, and Z. Wu, "Multi-strategy serial cuckoo search algorithm for global optimization," *Knowledge-Based Systems*, vol. 214, Article ID 106729, 2021.

[51] H. Peng, W. Zhu, C. Deng, and Z. Wu, "Enhancing firefly algorithm with courtship learning," *Information Sciences*, vol. 543, pp. 18–42, 2021.

[52] R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, "Self adaptive cuckoo search: analysis and experimentation," *Swarm and Evolutionary Computation*, vol. 60, Article ID 100751, 2021.

[53] R. Salgotra, U. Singh, S. Saha, and A. H. Gandomi, "Improving cuckoo search: incorporating changes for CEC 2017 and CEC 2020 benchmark problems," in *Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2020.

[54] M. Islam, P. Luo, and J. Sattar, "Simultaneous enhancement and super-resolution of underwater imagery for improved visual perception," 2020, https://www.semanticscholar.org/paper/Simultaneous-Enhancement-and-Super-Resolution-of-Islam-Luo.

[55] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[56] G. Chen, L. Dong, C. Dong, and X. Chen, "Image segmentation based on logistic regression sparrow algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 1-14, 2021.

[57] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Systems with Applications*, vol. 83, pp. 242–256, 2017.

[58] P. Upadhyay and J. K. Chhabra, "Kapur's entropy based optimal multilevel image segmentation using Crow Search Algorithm," *Applied Soft Computing*, vol. 97, Article ID 105522, Glasgow, UK, 2020.

[59] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2475460, 21 pages
https://doi.org/10.1155/2022/2475460

Research Article

A Multistrategy-Integrated Learning Sparrow Search Algorithm and Optimization of Engineering Problems

Zikai Wang, Xueyu Huang, and Donglin Zhu

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China

Correspondence should be addressed to Xueyu Huang; 6120190561@mail.jxust.edu.cn

Received 2 December 2021; Revised 15 January 2022; Accepted 26 January 2022; Published 23 February 2022

Academic Editor: Ripon Chakrabortty

The swarm intelligence algorithm is a new technology proposed by researchers inspired by the biological behavior of nature, which has been practically applied in various fields. As a kind of swarm intelligence algorithm, the newly proposed sparrow search algorithm has attracted extensive attention due to its strong optimization ability. Aiming at the problem that it is easy to fall into local optimum, this paper proposes an improved sparrow search algorithm (IHSSA) that combines infinitely folded iterative chaotic mapping (ICMIC) and hybrid reverse learning strategy. In the population initialization stage, the improved ICMIC strategy is combined to increase the distribution breadth of the population and improve the quality of the initial solution. In the finder update stage, a reverse learning strategy based on the lens imaging principle is utilized to update the group of discoverers with high fitness, while the generalized reverse learning strategy is used to update the current global worst solution in the joiner update stage. To balance exploration and exploitation capabilities, crossover strategy is joined to update scout positions. 14 common test functions are selected for experiments, and the Wilcoxon rank sum test method is achieved to verify the effect of the algorithm, which proves that IHSSA has higher accuracy and better convergence performance to obtain solutions than 9 algorithms such as WOA, GWO, PSO, TLBO, and SSA variants. Finally, the IHSSA algorithm is applied to three constrained engineering optimization problems, and satisfactory results are held, which proves the effectiveness and feasibility of the improved algorithm.

1. Introduction

In recent years, new intelligent optimization algorithms have emerged continuously and have been practically applied in medical treatment [1, 2], finance [3], production scheduling [4], and other fields. Besides, it has been proved to be remarkably effective. Since the end of the last century, scholars from all over the world have been inspired by social behavior [5], trying to simulate the behavior characteristics of biological populations in nature, and proposed algorithms such as Ant Colony Algorithm (ACO) [6, 7], Particle Swarm Optimization (PSO) [8, 9], Whale Optimization Algorithm (WOA) [10], Grey Wolf Optimization Algorithm (GWO) [11], and a series of swarm intelligence optimization algorithms. Most of the modeling process of these algorithms is based on the characteristics of the biological population, such as foraging [12], reproduction [13], and hunting [14], which vividly simulate the main behaviors in social actions. In 2020, Xue J and Shen B jointly proposed the Sparrow Search Algorithm (SSA) [1] based on the foraging behavior and backfeeding behavior of sparrow populations. The formula and control parameters of algorithm are not complex and are easier to be understood and implemented relatively. Experiments show that SSA's optimization capability is stronger than the particle swarm optimization algorithm proposed in 1995 and the grey wolf algorithm proposed in 2014, with better convergence accuracy, faster convergence speed, and better stability. However, compared with the existing swarm intelligence optimization algorithm, the SSA also has certain shortcomings, such as longer running time,

and a greater possibility to fall into a local optimal solution due to the excessively fast convergence speed, so that the global optimization ability is insufficient.

In order to strengthen the optimization effect of the algorithm and balance the capabilities of exploration and mining, researchers have proposed a series of improved methods for the original sparrow search algorithm model to improve the problem of easily trapping into local optimality. Of course, these improved methods of the swarm intelligence optimization algorithm have been used in various research fields, and the application has been realized extensively. Lv et al. [2] introduced a chaotic sequence to chaotically perturb some individuals, which fell into the local optimum, in order for the SSA to be jumped out of the restriction and continue to search for the global optimum solution. At the same time, the integrated Cauchy-Gaussian mutation operator is combined together to avoid the stagnation of optimization by changing the position of the elite sparrow in the search space. Zhu [15] introduced an adaptive learning factor to solve the problem that the convergence trend will slow down and the convergence accuracy will be reduced under a limited number of iterations, which are the shortcomings of the SSA. At the same time, the ASSA is applied to the optimization and identification of PEMFC stack parameters. Mao and Zhang [16] fused the sine cosine algorithm and the Levy flight strategy on the basic SSA, performed disturbance mutation at the optimal solution position, which enhanced the ability of the algorithm to escape locally, and greatly increased the accuracy of the solution. Liu et al. [17] and others introduced an improved sparrow search strategy to apply to the route planning problem of UAVs, which solved the inefficiency of path planning in the complex three-dimensional flight process. Yuan et al. [18] utilized the center of gravity reverse learning mechanism to initialize the population, which made the population distribution wider. A learning factor is put forward in the update part of the discoverer, and the mutation operator is introduced to increase the mutation processing and reduce the probability of the algorithm falling into the local optimum. Applying it to Distributed Maximum Power Point Tracking (DMPPT) provides conditions for the stable operation of the microgrid. Liu et al. [19] came up with a balanced sparrow search algorithm (BSSA), and the random walk strategy of Levy flight method was exerted to appropriately adjust the local search, which brought the improving efficiency of CNN focus. Besides, they applied it to the medical field to improve MRI image diagnosis of the brain robustness and accuracy of tumors. In order to solve the problem of labeled data classification, Zhang et al. [20] adopted the method of combining the improved SSA and the adaptive classifier and introduced the sine-cosine algorithm and the newly proposed labor cooperation structure. Great effect of application in the classification of lung CT images has been demonstrated. Zhang and Ding [21] designed a random configuration network based on the chaotic sparrow search algorithm, and, combined with the adaptive control factor of CSSA, it automatically updated the regularization parameters and scale factors for SCN. Thereby, the regression performance of SCN got improved when solving large-scale random configuration problems. Zhu and Yousefi [15] proposed to hold the adaptive sparrow search algorithm ASSA to optimize the seven unknown parameters of the proton exchange membrane fuel cell model in the PEMFC stack. The ultimate goal is achieving the best consistency with the empirical voltage polarization curve of the battery pack. Zhou et al. [22] successfully applied SSA to wavefront shaping and focusing by introducing a cross strategy, which solved the problem of SSA's lack of performance in high-dimensional optimization problems. Without a doubt, the improved algorithm provided a good reference for future wavefront shaping research.

Up to now, owing to the fact that the sparrow search algorithm has not been come up with for a long time, researchers are still in the exploratory stage and have not been able to develop an absolutely excellent algorithm. In order to further improve the solution accuracy and convergence efficiency of the sparrow algorithm, this paper continues to explore paths that can be improved on the basis of the predecessors and proposes the novel sparrow search algorithm called IHSSA. Improved infinite folding iterative chaotic mapping and hybrid reverse learning strategy are combined with it. The innovation points can be summarized as follows:

(1) The improved infinite folding iterative chaotic map (IICMIC) is used to initialize the sparrow population. This strengthens the diversity of the initial population to a certain extent and increases the breadth of distribution.

(2) A hybrid reverse learning strategy is put forward to update the position of a specific individual. Taking into account the effectiveness of reverse learning in mining new solutions, this paper uses a hybrid reverse learning strategy. After the discoverer is updated, lens reverse learning can be utilized to update the global optimal solution. After the position of the joiner is updated, the generalized opposition-based learning strategy contributes to update the current worst individual. Besides, considering the limitation of the boundary, the population can get more feasible areas as possible so as to maximize the mining.

(3) The horizontal and vertical crossing strategy is introduced to update the position of the guard. The advantage of this strategy is that it can update the individual sparrows in both the horizonal and vertical angles, while maintaining the solution speed, and the range of the population can be expanded to a certain extent.

This paper follows a reasonable logical order. The first chapter introduces the research background of intelligent algorithms in recent years and some contributions made by researchers to this field. The second chapter introduces the basic sparrow search algorithm SSA. The third chapter introduces several improvement points of this paper in sequence according to the application order, shows the

proposed new algorithm IHSSA, and attaches the flow chart of the new algorithm. The advantage of the algorithm is proved by time complexity analysis and Wilcoxon rank sum test, and the population distribution diagram proves its contribution to the dispersed population. In the following chapters, the new algorithm is tested on 14 standard test functions, the results are statistically tabulated, and a comparative analysis is made according to the data to verify the pros and cons of the algorithm. In Chapter 5, we apply IHSSA to a classical constraint engineering optimization problem, and the obtained data further proves the feasibility and effectiveness of the proposed algorithm. Finally, a brief summary of the work of this paper is made, and the author and his team have made some plans and prospects for the next research work.

2. Sparrow Search Algorithm SSA

2.1. Group Predation Behavior of Sparrows. In nature, as one of the common birds, sparrows live in the environment where humans live. Generally speaking, the upper body of the sparrow is brown and black, and the conical mouth is short and strong. They usually live together in groups with a clear division of labor. Some sparrows are responsible for finding food and providing foraging areas and directions for the entire population, while the remaining sparrows obtain food based on the food information the former sparrows provide. In addition, a sparrow in the population will issue an alarm in time when it realizes that danger is coming, and the entire population will quickly start backfeeding behavior.

2.2. SSA Algorithm Description. The proposal of SSA is based on the characteristics of sparrows' cleverness and strong memory, which well simulates the cooperative mechanism of sparrow populations in daily foraging. We will give new names to the three types of sparrows mentioned earlier. ① Those who are responsible for finding food are called discoverers. ② Those who follow the discoverers to obtain food are called joiners. ③ Some joiners will always monitor the discoverers and choose the time to compete for food resources in order to increase the rate of food acquisition. This type of joiner is called monitor. The discoverer generally accounts for 10%–20% of the entire population. The roles of the discoverer and the joiner can be exchanged, provided that the proportion relative to the entire population is constant.

The position of each sparrow is held as a solution of the algorithm. The initial positions of the sparrow represented by a matrix are as follows:

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}. \tag{1}$$

Among them, d represents the dimension of the problem to be optimized, and n represents the number of sparrow population. And then, the fitness value of all sparrows can be expressed:

$$F(x) = \begin{bmatrix} f\left(\left[x_{1,1}\ x_{1,2}\ \cdots\ x_{1,d}\right]\right) \\ f\left(\left[x_{2,1}\ x_{2,2}\ \cdots\ x_{2,d}\right]\right) \\ \vdots \\ f\left(\left[x_{n,1}\ x_{n,2}\ \cdots\ x_{n,d}\right]\right) \end{bmatrix}. \tag{2}$$

Among them, the function f represents the fitness function. The discoverer with better fitness will obtain food earlier in the food search process.

Since the discoverer needs to guide the foraging direction for the entire population, the discoverer can obtain a larger food search range. In the iterative process, the location of the discoverer is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \mathrm{EXP}\left(-\dfrac{i}{\alpha \cdot \mathrm{Maxitem}}\right), & R_2 < ST, \\ X_{i,j}^{t} + Q \cdot L, & R_2 \geq ST. \end{cases} \tag{3}$$

Among them, $X_{i,j}^{t}$ represents the current position of the i^{th} sparrow; *Maxitem* represents the maximum number of iterations of the algorithm; t is the current iteration number; α is a uniform number conforming to (0,1]; Q is a random number that obeys the standard normal distribution; L is a $1 * d$ matrix with each element being 1; alarm value $R_2 \in [0, 1]$; safety value $ST \in [0.5, 1]$. Once a sparrow in the population finds a predator or other danger, an alarm signal will be issued. When the alarm value is more than safety critical value, the discoverer will lead the population to other safer areas to forage.

The formula updated position of the follower is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \mathrm{EXP}\left(\dfrac{X_{\mathrm{worst}} - X_{i,j}^{t}}{i^2}\right), & i > \dfrac{n}{2}, \\ X_{b,j}^{t+1} + \left|X_{i,j}^{t} - X_{b,j}^{t+1}\right| \cdot A^{+} \cdot L, & \text{else.} \end{cases} \tag{4}$$

Among them, X_{worst} represents the current global worst position; $X_{b,j}^{t+1}$ represents the global optimal value of the j^{th} dimension at the $(t+1)^{\text{th}}$ iteration (that is, the best position of the discoverer); A represents a $1*d$ matrix whose elements are randomly assigned 1 or -1, A^{+} Satisfy $A^{+} = A^{T}(AA^{T})^{-1}$.

When $i > n/2$, it indicates that the i^{th} joiner has a low fitness level and is not able to obtain food. In order to obtain food and increase energy reserves, one must fly to other places for foraging; when $i \leq n/2$, it means that the i^{th} joiner has held the best position and randomly finds a location to forage near $X_{b,j}$.

The sparrows responsible for investigation generally account for 10%–20% of the total, and they always monitor and remind the entire population to take backfeeding behavior when facing danger. The position update formula of the monitors is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} + \beta \left| X_{i,j}^{t} - X_{b,j}^{t} \right|, & \mathrm{f}_i \neq f_g, \\ X_{i,j}^{t} + K \dfrac{X_{i,j}^{t} - X_{\mathrm{worst},j}^{t}}{\left(f_i - f_{\mathrm{worst}}\right) + \varepsilon}, & \mathrm{f}_i = f_g. \end{cases} \tag{5}$$

Among them, K is also a random number, and the range is $[-1, 1]$; ε is an infinitesimal constant, and its existence avoids the situation where the denominator is 0; f_i, f_g and f_{worst} represent the current fitness, the global optimal, and the global worst fitness value of the sparrow, respectively.

3. IHSSA

3.1. Infinitely Folded Iterative Chaotic Map Initialization Population

3.1.1. ICMIC. The swarm intelligence algorithm needs an initialization strategy to generate an initial population and provide an initial guess for the subsequent evolution process. The difference in the initial distribution state of the sparrow population will lead the entire subsequent foraging process to the final result with a large gap. Both the convergence speed and the optimization accuracy are deeply affected. Therefore, the importance of the quality of the initial population can be realized. According to the original SSA, the population is not guided by prior knowledge; that is, it is generally randomly generated. In 1975, Li et al. proposed the concept of "chaos" for the first time in the article "Period Three Implies Chaos" and used the word chaos for the first time [23].

Considering its unpredictability, ergodicity, and parameter sensitivity, chaotic systems are special. In the field of parameter optimization, chaotic mapping can be operated to replace pseudorandom number generators in order to generate chaotic numbers between 0 and 1. Considering that chaos can only traverse all the space in a sufficient length of time, it is feasible to combine chaos into the global optimizer to improve the search performance of the latter in order to complete the optimization of the target task in a short time range [24]. Experiments have proved that the utilization of chaotic sequences for population initialization will affect the entire process of the algorithm, and better results than pseudorandom numbers can often be held. The ergodicity of chaos allows the initial state of the sparrow population to have better diversity, to avoid premature convergence, that is, to improve the global optimization accuracy and convergence, which overcomes the shortcomings of traditional optimization algorithms.

This paper applies ICMIC map, one of the most classic chaotic maps (Iterative Chaotic Map with Infinite Collapses) to initialize the sparrow population. The chaotic map was proposed in 2001 by Di He. Its basic idea is to generate a chaotic sequence in $[0, 1]$ through the mapping relationship and then transform the chaotic sequence into the search space of the population [25]. Its higher Lyapunov exponent shows stronger chaotic characteristics than other commonly used continuous chaotic models [26]. Selecting appropriate parameters can generate a good chaotic model so as to contribute satisfactory results in practical applications. The uniform distribution test of chaotic systems by Di et al. [26] proved that the one-dimensional ICMIC presents a noise phenomenon closer to uniform distribution. Two mathematical expressions for ICMIC mapping are as follows:

Expression one:

$$\begin{aligned} x_{n+1} &= \sin\left(\frac{a}{x_n}\right), \\ -1 &\le x_n \le +1, \\ x_0 &\neq 0, \\ a &\in (0, +\infty), \\ \mathrm{n} &= 0, 1, 2, \ldots. \end{aligned} \tag{6}$$

Expression two:

$$\begin{aligned} x_{n+1} &= \sin\left(\frac{\alpha\pi}{x_n}\right) \\ \alpha &\in (0, 1). \end{aligned} \tag{7}$$

In expression one, a is a very important adjustable parameter. Experiments show that the value of a directly affects the mapping effect and then affects the pros and cons of the population. In the second expression, α also plays an important role as a control parameter.

3.1.2. IICMIC. Based on the expression two in 2.1.1, this paper proposes an improved infinite fold iterative chaotic map-IICMIC. The mathematical expression is as follows:

$$\begin{aligned} x_{n+1} &= \sin\left(\frac{\alpha\beta}{x_n}\right) \\ \beta &= 3\cos\left(r\mathrm{and}\left(1\right)\right) + \frac{1}{2}. \end{aligned} \tag{8}$$

After a lot of experiments, it is concluded that SSA can obtain a good chaotic sequence when the value of a is in the range of (0.6, 1). Combining IICMIC with the original SSA, the initial population state generated is shown in Figure 1(a) which shows the population state distribution after initial SSA initialization, and Figure 1(b) shows the distribution of sparrow population state after initialization with IICMIC. It can be seen that the improved initialization method has greatly improved the diversity of the population, and it has greatly avoided falling into the local optimum. The value of a is set 0.9 in subsequent experiments afterwards.

In combination with SSA, we first select initial values whose number is N with small differences as the initial state of the population. Taking into account the parameter sensitivity of ICMIC mapping, even if the individual gap is small, it can be captured. These N initial values can be mapped to get the same amount of chaotic sequence and then inversely mapped to the corresponding individual search space. The initial position of the i-th individual after the change is denoted as X_{n+1}^{i} ($i = 1.2, \ldots, D$).

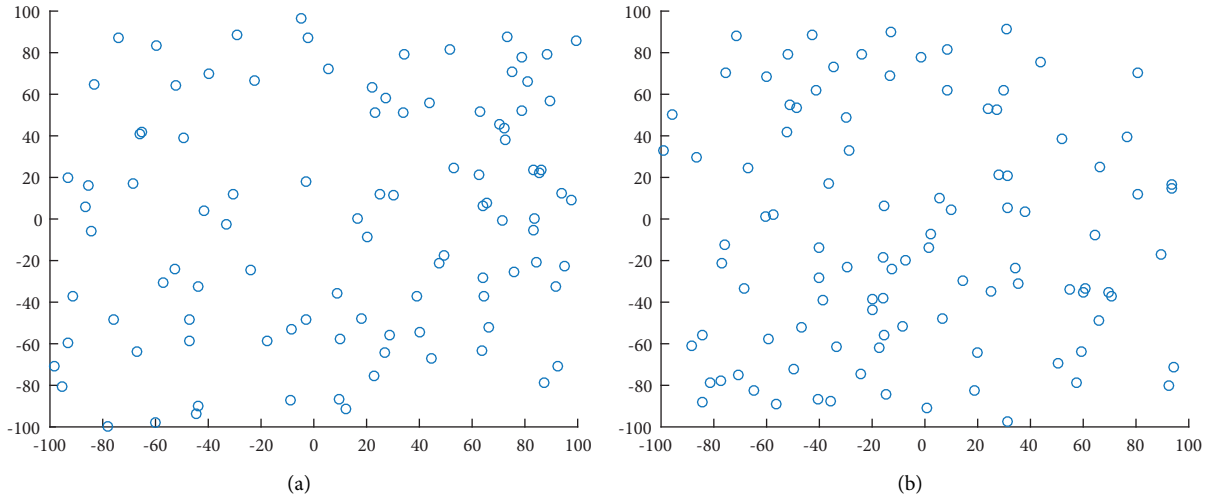


Figure 1: Individual distribution. (a) Individual initialization map of SSA. (b) Individual distribution of IHSSA.

3.2. Hybrid Reverse Learning Strategy to Update the Position of the Discoverer. Opposition-based learning (OBL) is an intelligent calculation method, which is first proposed by Tizhoosh in 2005. With the in-depth research of various algorithms, OBL has been successfully applied to many intelligent algorithms [27–31]. The main idea can be summarized as follows: calculate a feasible solution and its reverse solution. Then, evaluate the pros and cons of the two, and select the required solution according to certain conditions. Research shows that the solution generated by reverse learning is better than the randomly generated solution, and the probability of reaching the optimal solution is higher. Therefore, OBL is a good method that is greatly suitable for mining new solutions in unknown fields, which increases the diversity of the population.

In the discoverer stage, a broad and flexible search mechanism is the key to guide the entire sparrow population to search for food and avoid tripping into danger. In order to better realize the lead role of discoverers, it takes researchers too much time to explore in these fields, and they put forward a series of improvement methods gradually. However, traditional learning strategies have limited ability to solve problems and can only achieve their goals in certain dimensions. In response to this problem, based on the traditional OBL, this paper proposes a hybrid reverse learning method. Not only the improved lens imaging inverse learning mechanism is applied to the update of the optimal solution in the discoverer stage, but also the generalized opposition-based learning is performed on the global worst solution. The higher optimization accuracy can be obtained in this hybrid way so as to avoid premature convergence.

3.2.1. Reverse Learning Strategy Based on Improved Lens Principle to Update Optimal Position. The reverse learning strategy based on the lens principle has strong flexibility and versatility. A strong ability to explore unknown areas and dig new solutions is another advantage. The principles of this method are as follows:

Supposing that there is an object P with a height of h, and X_p is the projection of P on the X axis. Define a_j and b_j to be the upper and lower bounds of the solution in the j-th dimension under the current algorithm. The midpoint of the upper and lower bounds is defined as the base point O, and a lens with focal length f is placed at this point. Through the lens imaging, an image P' different from P can be obtained. The projection of the image P' on the X axis is denoted as X_P'. X_P' is the newly generated reverse solution based on this learning strategy. The schematic diagram is shown in Figure 2.

From Figure 2, we can clearly see that X_p generates a new image X_P' under the action of the lens. According to the properties of similar triangles, we can get the following formula:

$$\frac{(a+b)/2 - X_p}{X_P' - \mathrm{a} + b/2} = \frac{h}{h'}. \tag{9}$$

Let $h/h' = k$ (k is the scale factor), and the mathematical expression of the reverse point X_P' can be written as

$$X_P' = \frac{a+b}{2} + \frac{a+b}{2k} - \frac{X_p}{k}. \tag{10}$$

When $k = 1$, we can get

$$X_P' = a + b - X_p. \tag{11}$$

The formula above is the general form of the reverse learning strategy, and the new individuals generated by this formula are fixed. Studies have shown that, for high-dimensional complex functions, new individuals with a fixed range have a certain probability of falling into a local optimum. In the later stage of the algorithm iteration, the optimal solution generated is usually very close to the optimal solution. In order to deal with the hidden danger, we can introduce a new operator k^*. Changing the scale factor k contributes to dynamically variable and new individuals. The

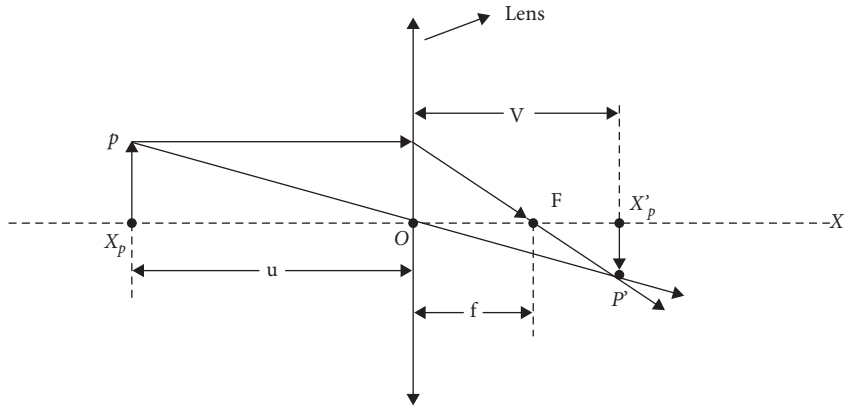


Figure 2: Lens schematic diagram.

randomness of the solution prevents individuals from losing vitality and increases the diversity of the population. The mathematical expression of k^* is as follows (the expression is implemented to the position of the i-th sparrow individual on the j-th dimension):

$$X_P^j = \frac{a^j + b^j}{2} + \frac{a^j + b^j}{2k^*} - \frac{X_{p,\mathrm{i}}^j}{k^*}$$
$$k^* = k + \sin\left(\pi \times \frac{T - t}{T}\right). \tag{12}$$

Among them, t represents the current number of iterations, and T represents the maximum number of iterations.

3.2.2. Generalized Reverse Learning Strategy Updates the Current Global Worst Position. With the deepening of research, more and more attempts have been made to choose the optimal solution, but the research on the current global worst position cannot be ignored. The update of the worst position can bring a better search range and make the population distribution greater. In order to maximize the diversity of individuals in the population, not only the best individuals in the discoverer stage are learned, but also the worst individuals in the sparrow group are learned in reverse strategy. Combined with the improved lens learning strategy in the previous subsection, this subsection adopts the generalized opposition-based learning (GOBL) strategy to optimize and update the current global worst position after each iteration.

The concept of generalized opposition-based learning is as follows: let the individual $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n})$, and their dynamic search range in the j-th dimension space is $[a_j, b_j]$. x_i' is its reverse solution. The mathematical expression of x_i' is

$$x_i' = k\left(a_j + b_j\right) - x_i. \tag{13}$$

Among them, k is a random number that obeys a uniform distribution between (0, 1). If the value range of the reverse solution exceeds the predetermined range, the solution will be randomly generated within the dynamic search range $[a_j, b_j]$ according to the following formula:

$$x_i' = a_j, b_j. \tag{14}$$

The purpose of reverse learning is to find a new and most suitable solution. Generalized opposition-based learning (GOBL) compares the worst solution while finding it and updates the current global worst solution once. At the same time, GOBL increases the dynamic update operation of the boundary than the basic reverse learning, which means the relatively small search space. The GOBL is combined with the worst joiner update of the sparrow search algorithm. The characteristics of reverse learning are fully utilized to explore more feasible regions while improving the convergence speed of the algorithm.

3.3. Vertical and Horizontal Cross Strategy. The optimization speed of the SSA is very fast, and the solution accuracy is also strong. As the number of iterations increases, the sparrow population will gather around the local optimal solution to a large extent. In order to balance the global search and development capabilities of the algorithm and avoid the algorithm from falling into the local optimum, the crossover optimization algorithm is newly proposed in 2014. It is inspired by the Confucian golden section principle and the crossover operation in genetic algorithms. The experimental results demonstrate that, compared with other heuristic algorithms, the cross-optimization algorithm has excellent performance on most test functions [32].

This paper proposes to integrate the vertical and horizontal crossover strategy into the guard search stage of the sparrow search algorithm, which expands the range of the population as much as possible while preserving the speed of the solution.

3.3.1. Horizontal Crossover Strategy. Horizontal crossover divides the solving space of multidimensional problem into half the population of hypercubes. In order to reduce the blind spots that cannot be reached, the horizontal crossover also searches the edge of each hypercube with a small probability. This is the guarantee that horizontal search has strong global search ability.

In this paper, two parental vigilant individuals $x_d^t(i)$ and $x_d^t(j)$ are crossed horizontally to generate new individuals $MSx_d^t(i)$ and $MSx_d^t(j)$.

$$MSx_d^t(i) = r_1 \times x_d^t(i) + (1 - r_1) \times x_d^t(j) + c_1 \times x_d^t(i) - x_d^t(j), \tag{15}$$

$$MSx_d^t(j) = r_2 \times x_d^t(j) + (1 - r_2) \times x_d^t(i) + c_2 \times x_d^t(j) - x_d^t(i). \tag{16}$$

Among them, r_1 and r_2 are random numbers in [0, 1] conforming to a uniform distribution, and c_1 and c_2 are random numbers in [−1, 1] conforming to a uniform distribution.

The offspring produced by the horizontal crossover needs to make an elite selection with their parents and retain the individuals with high adaptability. In this way, the algorithm can continuously converge to the optimal solution, which can ensure the convergence efficiency without affecting the optimization accuracy.

3.3.2. Vertical Cross Strategy. The premature convergence of most swarm intelligence search algorithms is caused by a small number of stagnant population dimensions. The original purpose of introducing vertical crossover is to promote certain dimensions of the population to escape from the dimensional convergence. Differing from the horizontal crossover strategy, the vertical crossover is operated on all dimensions of the new individual. Its function is to avoid premature maturity in the later stage of SSA, which is similar to the mutation mechanism in genetic algorithm.

Assuming that there is a newborn individual $x_d^t(k)$, which crosses longitudinally in the d_1 and d_2 dimensions, the calculation formula is as follows:

$$MSx_d^t(k) = r \times x_d^t(k) + (1 - r) \times x_d^t(k). \tag{17}$$

Among them, $MSx_d^t(k)$ is a new individual generated after vertical crossover, $r \in [0, 1]$.

Like the individuals generated by the horizontal crossover strategy, the new individuals generated after the vertical crossover must be selected by elites with their parents. The one with high adaptability is retained as the final individual. The advantage of this is that it not only increases the possibility of seeking the best in breadth, but also chooses various dimensions and realizes the continuous improvement of the quality of the solution. Even individuals who have fallen into a local optimum have a chance to jump out.

It is not difficult to see that, after combining with the horizontal and vertical crossover, it is indeed possible to balance the exploration and mining capabilities of the algorithm to a certain extent. The bottleneck in the horizontal direction can be shifted from the vertical experiment, and the vertical gains will be immediately fed back to the horizontal cross. Then, the information will spread to the entire population. The perfect combination of the two is like a layer of mesh structure that provides maximum help for optimization.

3.4. Frame Work of IHSSA. In summary, in order to solve the problems of the original sparrow search algorithm, such as fast convergence speed and high accuracy, but easy to mature early, several improvement measures have been proposed. The improved ICMIC is applied in the initialization phase, and the hybrid reverse learning strategy is utilized to update the discoverer and joiner, respectively. At the same time, the vertical and horizontal crossover strategy is added in the monitor stage to realize the overall update of each stage and strive to maximize the optimization. The specific implementation steps are as follows:

Step 1. Initialize the population and its parameters, including the population size *N*, the proportion of discoverers *PD*, the proportion of guards *SD*, the dimension of the objective function set to *D*, the upper and lower bounds of the initial value set to *lb* and *ub*, the maximum number of iterations *T*, and the alarm threshold *ST*, solving accuracy *ε*.

Step 2. Employ IICMIC to initialize the population (8), generate N D-dimensional vectors Zi, and then inversely map to the corresponding individual search space. The renewal of the population ensures the diversity of the sparrow population.

Step 3. Calculate the fitness f_i of each sparrow, select the current optimal fitness fb and its corresponding position xb, and the current worst fitness fw and its corresponding position xw.

Step 4. According to the set ratio PD, randomly select pNum sparrows with excellent adaptability as discoverers, and the rest become joiners. Update the position of the discoverers according to formula (3).

Step 5. According to the population fitness updated by the discoverer, an improved lens-based reverse learning strategy (12) is utilized to update the optimal value.

Step 6. Update the position of the joiner according to formula (4).

Step 7. Employ the generalized opposition-based learning strategy (13) to update the current global worst value.

Step 8. Randomly generate sNum guards from the population according to the ratio SD, and perform the horizontal crossover (15) and (16) operation.

Step 9. Perform vertical crossover operation according to formula (17), compare the degree of fitness, and save the better ones.

Step 10. According to the current state of the sparrow population, update the optimal position xb, the best fitness value fb, the worst position xw, and the worst fitness value fw of the entire population during the entire foraging process.

Step 11. Determine whether the iteration is over. If the algorithm reaches the maximum number of iterations, or the solution accuracy reaches the set value, it is determined that the loop ends, and the optimization result is output. Otherwise, it returns to Step2 to continue the next iteration operation, and the current iteration number t satisfies $t = t + 1$.

Step 12. Output the results of IHSSA.

The flow chart is shown in Figure 3.

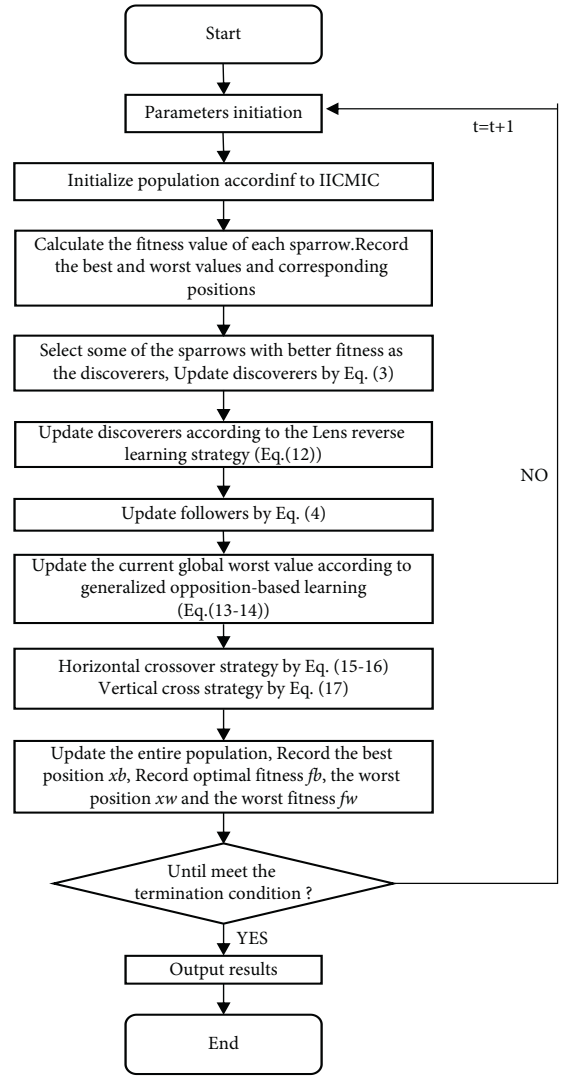


Figure 3: IHSSA flow chart.

4. Experimental Results and Analysis

4.1. Benchmark Function Test. In order to better verify the effectiveness of the newly improved algorithm, this paper selects 14 internationally representative benchmark functions for testing. The selected benchmark functions, which hold the function name, expression, and search interval of the function, are shown in Table 1. F1–F4 in the table are unimodal functions, usually only a global optimal value, the purpose of which is to test the local mining capability of the function. F5–F7 are multimodal functions which test the balance between exploration and mining of the function. The final selections F8–F14 are all fixed-dimensional functions. The theoretical optimal values of the 14 selected test functions are all 0.

All the algorithms mentioned are performed on Windows10 64 bit system, and the processor is Intel(R) Core(TM) i5-9300H CPU @ 2.40 GHz with 16 GB RAM. And the MATLAB R2016b simulation experiment platform is used for simulation.

4.2. Ablation Experiment. In order to verify the influence of the three improvement points of the algorithm on the effect of the entire experiment, an ablation experiment is hereby carried out. The comparison results are analyzed and have strong persuasion. The functions used for verification still select the 14 functions selected in the previous section, and the statistical results are divided into 5 angles according to the type and number of improvement points. These algorithms include the original SSA; the improved ICMIC initial population combined with the initial SSA is named ISSA-I; the improved algorithm combined the hybrid reverse learning strategy with the ISSA-I is named ISSA-II; the improved algorithm combined the crisscross strategy with ISSA-I is named ISSA-III; the last one is the IHSSA that combines all the innovations proposed in this paper. Integrate the data into Table 2 according to the principles above.

It can be seen from Table 2 that, in the process of improvement, the indicators of 8 functions have no obvious changes in the data. Among them, each index of the 7 functions, F1, F5, F6, F7, F9, F11, F12, and F13, reaches the optimal value of 0 in the SSA. The value obtained by the improved algorithm still keeps the optimal state. With the increase of improvement points, in the five functions of F2, F3, F4, F10, and F14, the optimization effect becomes more significant. Except for the best optimization of ISSA-III in F14, the other four functions are all the best optimization values of IHSSA and even get progress of many orders of magnitude. In F8, although there is no improvement in the two data of average and standard deviation, the optimal value has reached an improvement of 7 orders of magnitude. Overall, the IHSSA that combines all the innovation points proposed in this article has the best effect. Each innovation point has played a certain role in each step of the algorithm; especially combining IICMIC with population initialization has brought obvious results.

4.3. Population Diversity Analysis. Population diversity is one of the important performance indexes to measure the pros and cons of an algorithm, which can reflect whether the algorithm falls into a local optimum to a certain extent. In this paper, the population distribution map in the early stage of the iteration (the number of iterations is 10) was selected as a reference. The unimodal function F1 and multimodal function F8 proposed in the above table were selected as the research objects to show the advantages and disadvantages of IHSSA and the original SSA, as shown in Figures 4(a) and

Table 1: Fourteen benchmark test functions.

Function name	Function	Dimension	Interval
Sphere	$F_1(\mathrm{x}) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$
Schwefel's problem 1.2	$F_2(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]$
Schwefel's problem 2.21	$F_3(x) = \max_i\{\lvert x_i\rvert, 1 \le \mathrm{i} \le n\}$	30	$[-100, 100]$
Rosenbrock	$F_4(x) = \sum_{\mathrm{i}=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2)]$	30	$[-30, 30]$
Rastrigin	$F_5(x) = \sum_{\mathrm{i}=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]$
Ackley	$F_6(x) = -20\exp(-0.2\sqrt{1/n}\sum_{i=1}^{n} x_i^2) - \exp((1/n)\sum_{\mathrm{i}=1}^{\mathrm{n}} \cos(2\pi x_i)) + 20 + e$	30	$[-32, 32]$
Griewank	$F_7(x) = 1/4000\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	30	$[-600, 600]$
Schwefel	$F_8(x) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i\rvert})$	30	$[-500, 500]$
Three-hump camel	$F_9(x) = 2x_1^2 - 1.05x_1^4 + x_1^6/6 + x_1x_2 + x_2^2$	2	$[-5, 5]$
Colville	$F_{10}(x) = 100x_1^2 - x_2^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	4	$[-10, 10]$
Bent cigar	$F_{11}(x) = x_1^2 + 10^6\sum_{i=2}^{n} x_i^2$	10	$[-10^{10}, 10^{10}]$
Zakharov	$F_{12}(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$	10	$[-5^{10}, 10^{10}]$
Noncontinuous rotated Rastrigin's	$F_{13}(x) = \sum_{i=1}^{n} (z_i^2 - 10\cos(2\pi z_i) + 10) + F_{13}^*$, $\widehat{X} = M_1 5.12(x - o)/100 y_i = \begin{cases} \widehat{X_i}, & if\ \lvert\widehat{X}_i\rvert \le 0.5, \\ \mathrm{round}(2\widehat{X}_i/2), & if\ \lvert\widehat{X}_i\rvert > 0.5, \end{cases}$ $Z = M_1\Lambda^{10}M_2T_{asy}^{0.2}(T_{osz}(y))$	10	$[-5^{10}, 5^{10}]$
Levy function	$F_{14}(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{n-1} (\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_1 + 1)] + (\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)]$, Where $\omega_i = 1 + x_i - 1/4$	30	$[-10^{30}, 10^{30}]$

Table 2: Algorithm parameters

Algorithm	Parameters
GWO	$a_{max} = 2$, $a_{min} = 0$
PSO	$c_1 = c_2 = 1.49445$
SSA	$PD = 0.2$, $ST = 0.6$, $SD = 0.2$
CSSA	$PD = 0.2$, $ST = 0.8$, $SD = 0.2$
LSSA	$PD = 0.2$, $SD = 0.2$
GSSA	$PD = 0.3$, $ST = 0.6$, $SD = 0.7$
YSSA	$PD = 0.2$, $SD = 0.2$
IHSSA	$PD = 0.2$, $SD = 0.2$

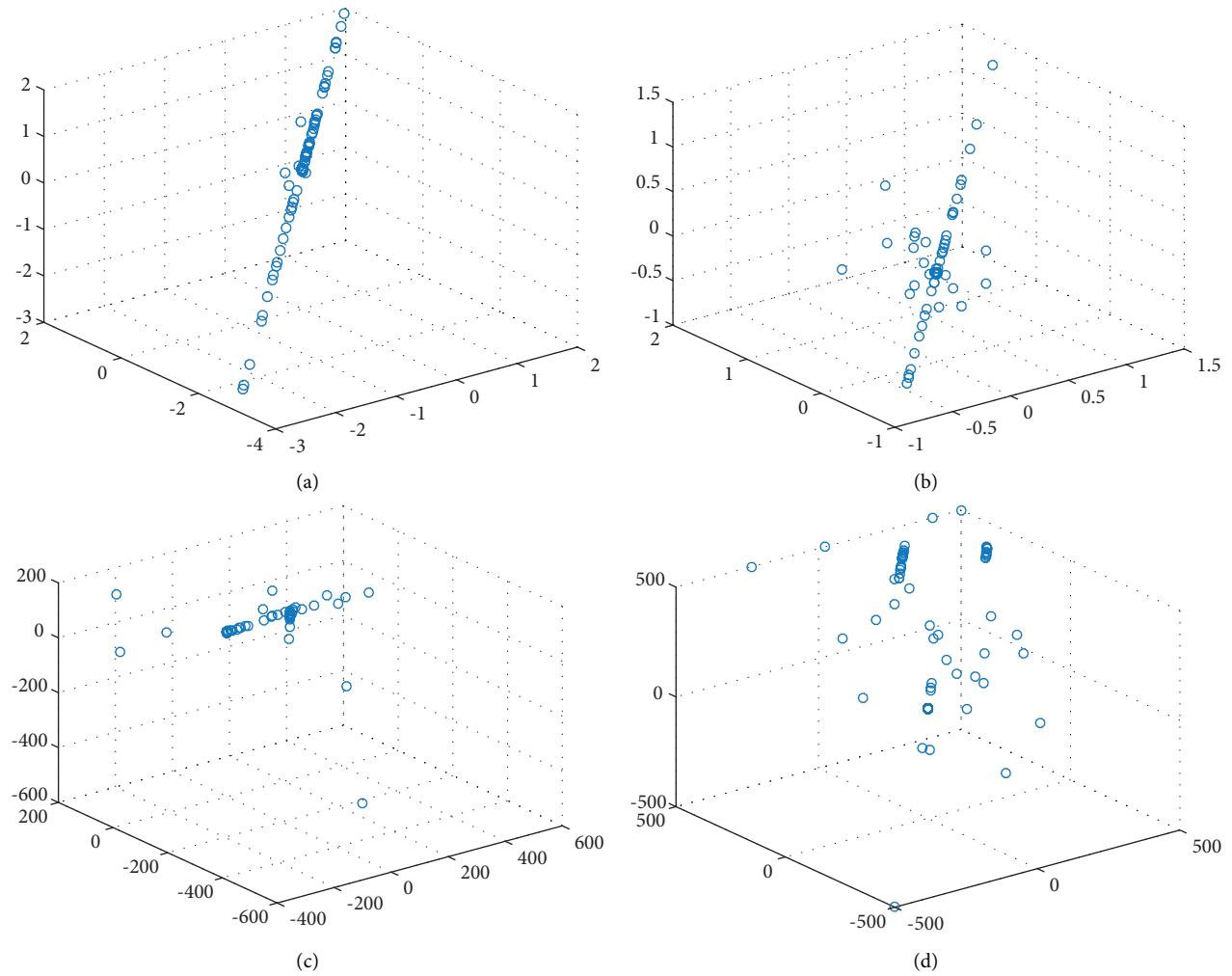


Figure 4: Population distribution map. (a) SSA. (b) IHSSA. (c) SSA. (d) IHSSA.

4(b), respectively, representing the individual distribution of SSA and IHSSA on F1, and Figures 4(c) and 4(d), respectively, represent the individual distribution of SSA and IHSSA on F8. The theoretical optimal value of F1 is 0, and the theoretical optimal value for F8 is 420.

As can be seen from Figure 4, in the early stage of the algorithm iteration, the distribution in Figure 4(a) is linear, while the IHSSA in Figure 4(b) is more widely distributed. Compared with the poor aggregation state of SSA in Figure 4(c), the distribution shown in Figure 4(d) is closer to the theoretical optimal value and presents a wider distribution field. It can be seen that the improved IHSSA in this paper increases the diversity of the population to a certain extent and reduces the invalid search of individuals.

4.4. Comparison with Other Optimization Algorithms. 14 standard test functions proposed in the previous section are utilized to test the performance of the improved IHSSA. Nine intelligent optimization algorithms, including particle swarm optimization (PSO), whale optimization algorithm (WOA), grey wolf optimization algorithm (GWO), teaching

and learning algorithm (TLBO), Sparrow Search Algorithm (SSA), Chaos Sparrow Optimization Algorithm (CSSA) proposed by Lv et al. [2], LSSA improved by Zhu DL [33], GSSA improved by Chen G, and YSSA proposed by Yan et al. [34], are chosen for comparison. In order to ensure the objectivity of the experiment and the fairness of comparison, the population size and maximum iteration number of each algorithm are 100 and 500, respectively. The other parameter settings of the 8 algorithms are shown in Table 3. Considering the importance of parameter values in experimental results, the feasibility proved by a large number of experiments is the only source of value, so the data in the table are from the parameter values set by the author when each algorithm was first proposed. In order to avoid the contingency of the algorithm results, each test function is run 30 times separately. And the average value, standard deviation and optimal value of the experiment are calculated, respectively. Meanwhile, the average running time of each algorithm for optimizing in each function was recorded as a reference for improving performance. The experimental data are shown in Table 4.

It can be seen from Table 4 that, compared with the other three SSA algorithms, the same results are achieved in 7 functions; even the 6 functions F1, F5, F7, F9, F11, and F13 have found the optimal solution 0. There are obvious improvements in the remaining 7 functions, and the average value of optimization in F2, F3, and F4 has been improved by multiple orders of magnitude. Compared with the WOA, GWO, and TLBO algorithms, the optimal solution 0 is found in F5, F9, and F7, respectively, and the results in the other functions are better. Compared with the PSO algorithm, the five functions of F4, F11, F12, F13, and F14 have a significant improvement, which is particularly prominent in F4. In addition, compared with the basic SSA, the optimal values found in the three functions are improved significantly. Compared with the other two improved SSA algorithms, the results are better in the three functions of F4, F10, and F14.

In F8, apart from the GSSA, the performance of several SSA algorithms is not as good as WOA, especially in the average value. Overall, the IHSSA proposed in this paper has the best performance among the 14 functions, while the PSO has the worst performance. Figure 5 shows the convergence curves of 8 algorithms for 10 functions. It can be seen that, among the five functions of F1, F2, F3, F6, and F12, IHSSA has the fastest convergence speed and higher convergence accuracy. In F4, F10, and F14, although IHSSA has the same convergence speed as other SSA variants, it is obviously able to obtain a better solution. For F8, WOA showed a high advantage, and GSSA shows superior optimization ability than other SSA variants. However, compared with SSA and LSSA, IHSSA performs better in convergence accuracy, but compared with GSSA, CSSA, and ISSA, the accuracy is still far from the theoretical optimal value. In terms of running time, the variant of SSA consumes more time than the original SSA. However, among several variants, LSSA and IHSSA have relatively shorter running times, and higher efficiency in the optimization process of 7 functions, respectively.

In general, IHSSA has the fastest convergence speed and better convergence accuracy; that is, the quality of the algorithm's optimal solution is better.

4.5. Wilcoxon Rank Sum Test. Derrac et al. proposed that, for the performance evaluation of improved intelligent optimization algorithms, data comparison only based on average, standard, and optimal values is not convincing enough. One of the necessary conditions, the quality of the statistical test results, also proves whether the algorithm has been significantly improved or not. In order to judge that the results of the improved IHSSA in this paper are significantly different from the results of other algorithms, the Wilcoxon statistical test was performed at a significance level of 5% [23]. The test principle is briefly described as follows: when $P < 0.05$, it is considered that there is a significant difference between the two algorithms. When $P < 0.05$, it indicates that the performance of the two algorithms is equivalent, and the difference is not obvious. In this article, the partial value of $P > 0.05$ is expressed as *N/A*. Table 5 shows the P value calculated in the Wilcoxon rank sum test of IHSSA and other algorithms among the 14 selected benchmark functions. The results show that $P < 0.05$ accounts for the main component. The IHSSA has a greater improvement over the SSA algorithm, and its superiority is also statistically significant, which proves that the improved algorithm has a higher convergence accuracy.

4.6. Time Complexity Analysis. Time complexity is one of the important indicators for judging the performance of the algorithm and calculating the running cost. Analyze whether the improved IHSSA increases the time complexity from both the macro- and microperspectives. On the one hand, from a macroperspective, supposing that the maximum number of iterations of the algorithm is M, the dimension is D, and the population size is P, then, according to the time complexity calculation formula of the intelligent optimization algorithm, the time complexity of SSA is $O_1 = P \times M \times D$. For the improved IHSSA, although the number of cycles has been increased, the structure of the algorithm has not changed. Therefore, the time complexity O_2 of the IHSSA can be calculated as $O_2 = P \times M \times D$. Obviously, $O_1 = O_2$, and the time complexity has not increased in the macroscopic view. On the other hand, from a microperspective, the time complexity of IHSSA has increased to a certain extent. Assuming that the proportions of discoverers and joiners are A and B, respectively, then, the time complexity of lens-based reverse learning O_3 and generalized opposition-based learning O_4 is $O_3 = A \times P \times M \times D$, $O_4 = M$, respectively. The increase in time complexity of the alert phase of the vertical and horizontal cross strategy update is $O_5 = B \times P \times M \times D$. The initialization phase of IICMIC does not increase the time complexity. In summary, from a microscopic point of view, the time complexity of the improved algorithm has increased by $O_t = O_3 + O_4 + O_5 = (A + B) \times P \times M \times D + M$, but the increase in each step did not cause orders of magnitude. The total time complexity is still $P \times M \times D$.

Table 3: Ablation experiment.

Function	Index	SSA	ISSA-I	ISSA-II	ISSA-III	IHSSA
F1	Avg	0	0	0	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F2	Avg	4.9592*E*-290	0	1.1301*E*-287	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F3	Avg	2.9952*E*-193	1.0566*E*-230	5.119*E*-163	1*E*-233	0
	Std	0	0	0	0	0
	Best	0	0	4.7959*E*-165	0	0
F4	Avg	1.3694*E*-05	8.13949*E*-06	2.59124*E*-05	4*E*-07	2*E*-07
	Std	3.77399*E*-05	2.05064*E*-05	5.66546*E*-05	2*E*-06	8*E*-07
	Best	1.36288*E*-09	1.88962*E*-10	2.45252*E*-10	7*E*-12	0
F5	Avg	0	0	0	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F6	Avg	8.88178*E*-16	8.88178*E*-16	8.88178*E*-16	9*E*-16	9*E*-16
	Std	0	0	0	0	0
	Best	8.88178*E*-16	8.88178*E*-16	8.88178*E*-16	9*E*-16	9*E*-16
F7	Avg	0	0	0	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F8	Avg	3958.182107	3972.12115	2628.086289	4674.8	3377.4
	Std	712.2420957	527.205402	1002.927817	770.11	1620.5
	Best	2389.812363	3085.587428	0.014612768	3396.7	0.0008
F9	Avg	0	0	0	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F10	Avg	4.82652*E*-08	3.93007*E*-08	1.53723*E*-08	6*E*-08	2*E*-08
	Std	1.02745*E*-07	8.62134*E*-08	3.49383*E*-08	2*E*-07	5*E*-08
	Best	2.87751*E*-13	2.76833*E*-14	2.82932*E*-20	2*E*-13	4*E*-15
F11	Avg	0	0	0	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F12	Avg	0	0	0	1*E*-188	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F13	Avg	0	0	0	0	0
	Std	0	0	0	0	0
	Best	0	0	0	0	0
F14	Avg	5.14521*E*-10	8.78436*E*-10	1.26547*E*-09	1*E*-11	2*E*-10
	Std	1.83095*E*-09	4.04952*E*-09	2.0902*E*-09	5*E*-11	8*E*-10
	Best	2.07524*E*-20	3.3079*E*-14	3.0117*E*-11	2*E*-15	2*E*-14

Table 4: Comparisons of IHSSA and other seven algorithms for 14 test functions.

Function	Algorithm	Avg	Std	Best	Run time
F1	WOA	6.47271*E*-97	2.45955*E*-96	1.4869*E*-104	0.00114
	GWO	9.58408*E*-41	1.61375*E*-40	1.85345*E*-42	0.001135
	PSO	4.51174*E*-11	6.25117*E*-11	1.31715*E*-12	0.003173933
	TLBO	2.29067*E*-85	1.79774*E*-85	2.6797*E*-86	0.0051
	SSA	0	0	0	0.001907
	CSSA	0	0	0	0.002318533
	LSSA	0	0	0	0.002167
	GSSA	3.826*E*-123	1.4763*E*-122	0	0.003030333
	YSSA	0	0	0	0.0024253
	IHSSA	0	0	0	0.00231

Table 4: Continued.

Function	Algorithm	Avg	Std	Best	Run time
F2	WOA	14907.66827	7290.613484	5245.501087	0.000753
	GWO	2.01843*E*-11	4.84067*E*-11	2.37886*E*-15	0.001054
	PSO	6.159248452	3.416561147	2.274329699	0.003053
	TLBO	1.49173*E*-15	1.26326*E*-15	1.28014*E*-16	0.005033
	SSA	4.9592*E*-290	0	0	0.001873
	CSSA	0	0	0	0.001958
	LSSA	0	0	0	0.002093
	GSSA	1.03442*E*-88	1.13315*E*-87	0	0.002994
	YSSA	0	0	0	0.00206
	IHSSA	0	0	0	0.001953
F3	WOA	34.68527033	29.6668357	4.83669*E*-05	0.000001
	GWO	2.5165*E*-10	2.80518*E*-10	3.92494*E*-11	0.000071
	PSO	0.210838877	0.155290345	0.057831908	0.002197
	TLBO	2.30203*E*-34	1.48485*E*-34	3.57945*E*-35	0.004141
	SSA	2.9952*E*-193	0	0	0.000924
	CSSA	0	0	0	0.00118
	LSSA	0	0	0	0.001134
	GSSA	4.87917*E*-90	2.20121*E*-89	0	0.002007
	YSSA	0	0	0	0.001297
	IHSSA	0	0	0	0.001171
F4	WOA	28.72457614	0.198079568	27.98805166	0.000752
	GWO	1.62822*E*+35	6.51934*E*+35	38250907192	0.001087
	PSO	3.24434*E*+87	4.65939*E*+86	2.29319*E*+87	0.002976
	TLBO	420.4775663	1327.016075	21.95591078	0.004966
	SSA	1.3694*E*-05	3.77399*E*-05	1.36288*E*-09	0.001724
	CSSA	6.06157*E*-06	1.48998*E*-05	1.03023*E*-08	0.001897
	LSSA	5.93493*E*-06	1.14239*E*-05	0	0.001974
	GSSA	5.11701*E*-07	5.04167*E*-07	1.6666*E*-14	0.002881
	YSSA	1.50157*E*-05	3.02452*E*-05	2.37868*E*-09	0.002006
	IHSSA	2.40169*E*-07	8.20733*E*-07	0	0.001887
F5	WOA	0	0	0	0.000587
	GWO	1.581667461	2.939452422	0	0.001086
	PSO	45.60347387	12.12604405	24.87396229	0.003016
	TLBO	6.383023506	4.955050004	0	0.005033
	SSA	0	0	0	0.001767
	CSSA	0	0	0	0.001891
	LSSA	0	0	0	0.001917
	GSSA	0	0	0	0.0028
	YSSA	0	0	0	0.002885
	IHSSA	0	0	0	0.00188
F6	WOA	5.15143*E*-15	2.16807*E*-15	8.88178*E*-16	0.000032
	GWO	2.6823*E*-14	3.63147*E*-15	1.86517*E*-14	0.000032
	PSO	0.343589201	0.596942294	1.24699*E*-06	0.002345
	TLBO	0.031044156	0.170035847	4.44089*E*-15	0.004333
	SSA	8.88178*E*-16	0	8.88178*E*-16	0.001045
	CSSA	8.88178*E*-16	0	8.88178*E*-16	0.001243
	LSSA	8.88178*E*-16	0	8.88178*E*-16	0.001302
	GSSA	8.88178*E*-16	1.00293*E*-31	8.88178*E*-16	0.002071
	YSSA	8.88178*E*-16	1.00293*E*-31	8.88178*E*-16	0.001365
	IHSSA	8.88178*E*-16	0	8.88178*E*-16	0.001232
F7	WOA	0.00420253	0.012960719	0	0.000702
	GWO	0.001780922	0.004128695	0	0.000986
	PSO	0.017048805	0.019184635	6.22012*E*-11	0.002873
	TLBO	0	0	0	0.004866
	SSA	0	0	0	0.001613
	CSSA	0	0	0	0.001893
	LSSA	0	0	0	0.001803
	GSSA	0	0	0	0.002723
	YSSA	0	0	0	0.001993
	IHSSA	0	0	0	0.001885

Table 4: Continued.

Function	Algorithm	Avg	Std	Best	Run time
F8	WOA	749.0614488	1053.51792	0.142228645	0.001887
	GWO	5984.325616	512.034691	5075.836656	0.001087
	PSO	5822.761982	781.2113436	3040.698318	0.002733
	TLBO	5050.58229	1189.147899	3306.384094	0.004777
	SSA	3958.182107	712.2420957	2389.812363	0.001487
	CSSA	1619.35044	977.2374638	217.1401425	0.001995
	LSSA	4878.962977	846.4505234	3517.371964	0.001683
	GSSA	678.1120161	1382.220669	0.000381827	0.002487
	YSSA	2178.636278	2045.618662	0.000381827	0.002057
	IHSSA	3377.438965	1620.549772	0.000815262	0.001987
F9	WOA	1.2342*E*-117	5.3288*E*-117	2.672*E*-142	0.000885
	GWO	0	0	0	0.001102
	PSO	7.15959*E*-48	2.6399*E*-47	6.0523*E*-53	0.002666
	TLBO	6.1741*E*-183	0	1.5723*E*-188	0.004666
	SSA	0	0	0	0.001424
	CSSA	0	0	0	0.001911
	LSSA	0	0	0	0.001614
	GSSA	0	0	0	0.002457
	YSSA	0	0	0	0.002011
	IHSSA	0	0	0	0.001902
F10	WOA	0.754881452	1.386931349	0.001459345	0.000757
	GWO	0.600742635	1.39373905	2.53554*E*-05	0.000965
	PSO	0.000892468	0.000922365	1.42838*E*-06	0.002883
	TLBO	3.78444*E*-06	9.24279*E*-06	3.07036*E*-08	0.004883
	SSA	4.82652*E*-08	1.02745*E*-07	2.87751*E*-13	0.001633
	CSSA	6.06071*E*-08	1.74794*E*-07	4.9792*E*-14	0.001969
	LSSA	1.87914*E*-06	3.82252*E*-06	0	0.001777
	GSSA	9.6035*E*-07	1.483*E*-06	2.07901*E*-28	0.002665
	YSSA	5.06037*E*-08	9.41366*E*-08	2.87751*E*-13	0.002075
	IHSSA	1.78222*E*-08	5.32296*E*-08	3.72464*E*-15	0.001965
F11	WOA	6.5603*E*-78	3.17192*E*-77	9.75895*E*-90	0.00132
	GWO	6.77957*E*-66	2.37196*E*-65	2.19507*E*-70	0.000106
	PSO	1.0187*E*+26	2.21398*E*+25	4.6921*E*+25	0.002255
	TLBO	4.82046*E*-85	5.52867*E*-85	1.68795*E*-86	0.004233
	SSA	0	0	0	0.000977
	CSSA	0	0	0	0.001212
	LSSA	0	0	0	0.001273
	GSSA	0	0	0	0.002087
	YSSA	0	0	0	0.001338
	IHSSA	0	0	0	0.001206
F12	WOA	6.6988*E*+16	9.54778*E*+16	39646.29751	0.000265
	GWO	2.61392*E*+15	3.73552*E*+15	1.47252*E*+14	0.000282
	PSO	2.38287*E*+43	1.23114*E*+43	4.17747*E*+42	0.002377
	TLBO	9.02546*E*+14	4.72246*E*+14	2.97075*E*+14	0.004306
	SSA	0	0	0	0.000983
	CSSA	0	0	0	0.001293
	LSSA	0	0	0	0.001166
	GSSA	0	0	0	0.001983
	YSSA	0	0	0	0.001412
	IHSSA	0	0	0	0.001282
F13	WOA	0.7	0	0	0.001367
	GWO	0.8	2.006884702	0	0.000958
	PSO	1.22453*E*+20	2.91996*E*+19	5.79476*E*+19	0.003133
	TLBO	4.87463304	0.988415075	2.761251106	0.005187
	SSA	0	0	0	0.001922
	CSSA	0	0	0	0.002064
	LSSA	0	0	0	0.002097
	GSSA	0	0	0	0.002965
	YSSA	0	0	0	0.002185
	IHSSA	0	0	0	0.002057

Table 4: Continued.

Function	Algorithm	Avg	Std	Best	Run time
F14	WOA	591670353.7	3240707887	1.967264854	0.000187
	GWO	5.22374*E*+34	1.32705*E*+35	5.04003*E*+32	0.000466
	PSO	1.94575*E*+60	2.11926*E*+59	1.42718*E*+60	0.002175
	TLBO	0.320134144	0.127010931	0.093171452	0.00516
	SSA	5.14521*E*-10	1.83095*E*-09	2.07524*E*-20	0.000983
	CSSA	3.47974*E*-10	7.18343*E*-10	9.49837*E*-14	0.001377
	LSSA	1.00461*E*-06	2.21303*E*-06	1.49976*E*-32	0.001183
	GSSA	1.45943*E*-07	2.58657*E*-07	7.98889*E*-19	0.002022
	YSSA	2.20337*E*-07	6.82432*E*-07	1.49976*E*-32	0.001485
	IHSSA	1.96838*E*-10	7.70225*E*-10	2.39969*E*-14	0.001365

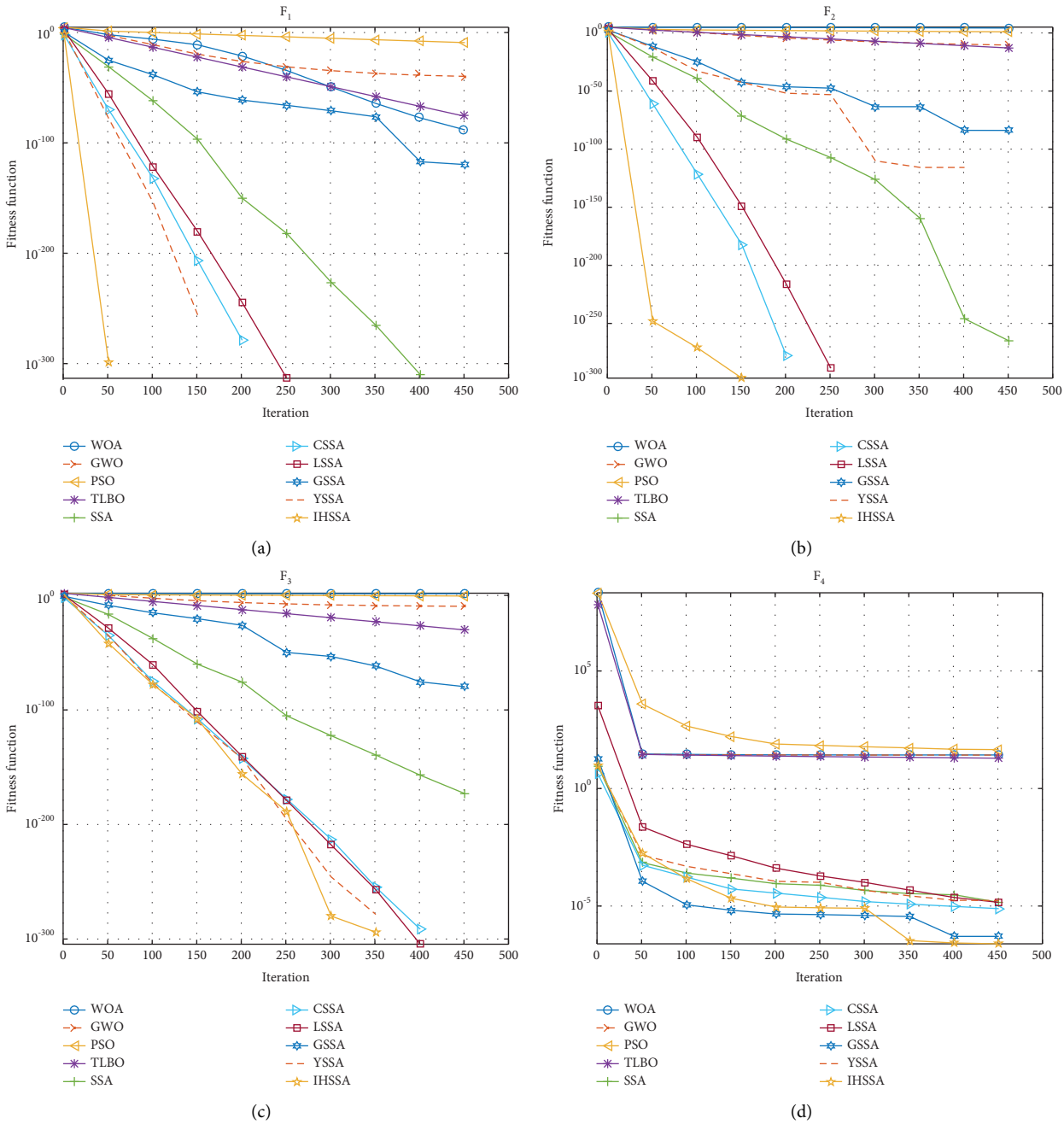
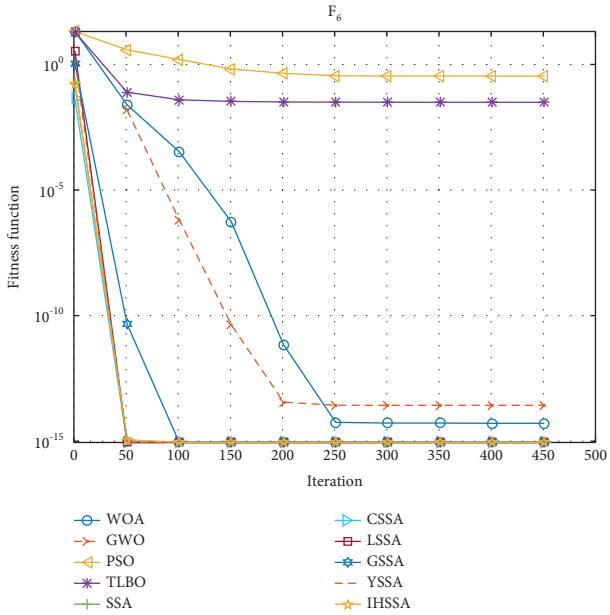
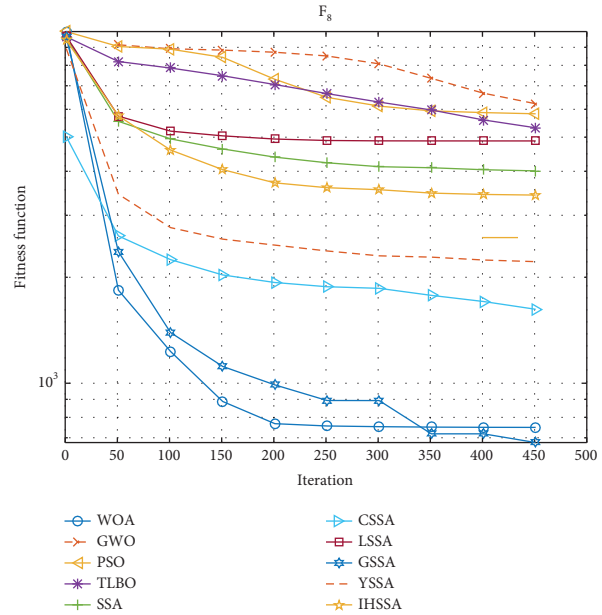


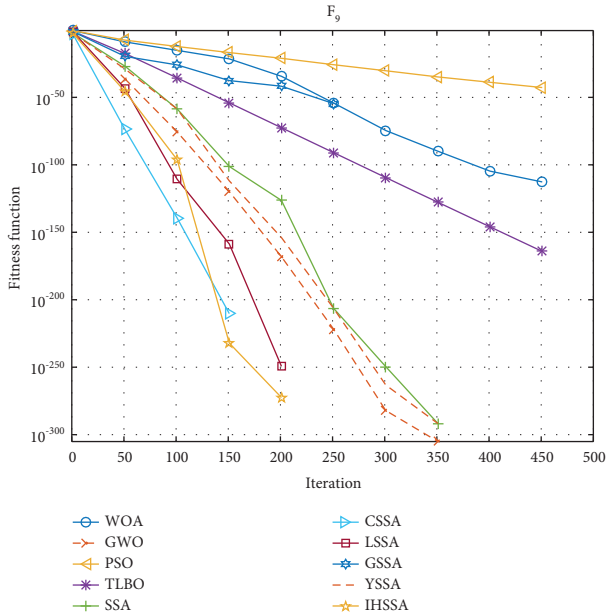
Figure 5: Continued.



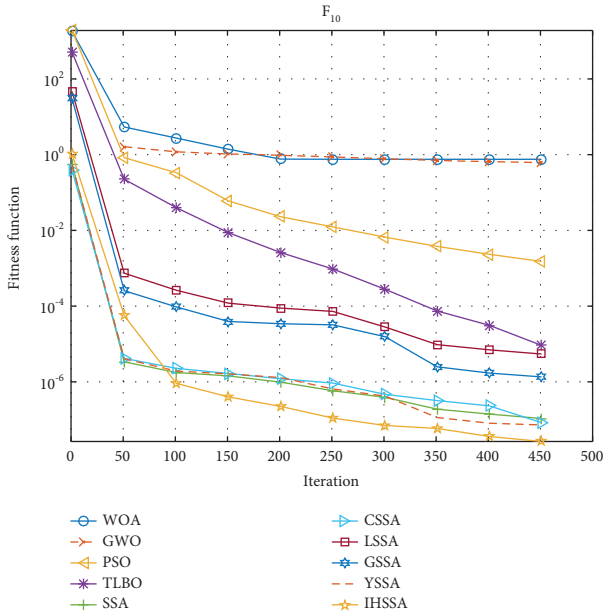
(e)



(f)



(g)



(h)

Figure 5: Continued.

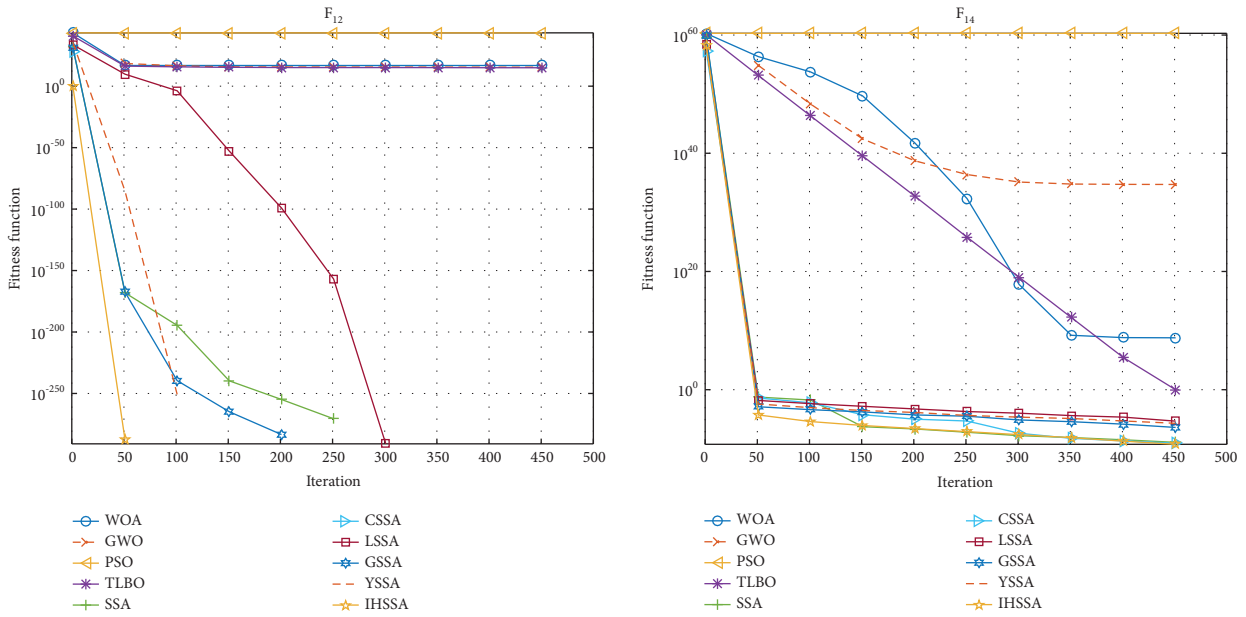


Figure 5: Convergence curves of eight algorithms for ten representatives test functions. *Note.* (a) corresponds to F1, (b) corresponds to F2, (c) corresponds to F3, (d) corresponds to F4, (e) corresponds to F6, (f) corresponds to F8, (g) corresponds to F9, (h) corresponds to F10, (i) corresponds to F12, and (j) corresponds to F14. The image optimization results of the four functions F5, F7, F11, and F13 have greater advantages and reach the optimal value after a short number of iterations. Since the convergence effect is too good, considering the overall beauty of the image, it will not be displayed.

Table 5: p value and Wilcoxon rank.

Function	WOA	GWO1	PSO1	TLBO	SSA	CSSA	LSSA	GSSA	YSSA
F1	3.02*E*-11	3.02*E*-11	3.02*E*-11	3.02*E*-11	N/A	N/A	0.049941793	N/A	0.006518796
F2	1.21*E*-12	1.21*E*-12	1.21*E*-12	1.21*E*-12	0.333710696	N/A	N/A	4.79*E*-08	N/A
F3	1.21*E*-12	1.21*E*-12	1.21*E*-12	1.21*E*-12	0.002788006	N/A	N/A	1.93*E*-10	N/A
F4	3.02*E*-11	3.02*E*-11	3.02*E*-11	3.02*E*-11	6.53*E*-08	9.26*E*-09	0.0239	9.13*E*-04	7.49*E*-08
F5	N/A	2.15*E*-06	1.21*E*-12	1.93*E*-10	N/A	N/A	N/A	N/A	N/A
F6	2.53*E*-11	5.67*E*-13	1.21*E*-12	3.50*E*-13	N/A	N/A	N/A	N/A	N/A
F7	0.081522972	0.021577192	1.21*E*-12	N/A	N/A	N/A	N/A	N/A	N/A
F8	1.49*E*-06	5.57*E*-10	4.18*E*-09	4.12*E*-06	0.02920541	3.09*E*-06	1.61*E*-06	6.51*E*-07	0.1259
F9	1.21*E*-12	N/A	1.21*E*-12	1.21*E*-12	N/A	N/A	N/A	N/A	N/A
F10	3.02*E*-11	3.02*E*-11	3.02*E*-11	1.46*E*-10	0.222572896	0.620403721	0.003475701	0.0199	0.0271
F11	1.21*E*-12	1.21*E*-12	1.21*E*-12	1.21*E*-12	N/A	N/A	N/A	N/A	N/A
F12	1.21*E*-12	1.21*E*-12	1.21*E*-12	1.21*E*-12	N/A	N/A	N/A	N/A	N/A
F13	0.0815	0.0028	1.21*E*-12	1.21*E*-12	N/A	N/A	N/A	N/A	N/A
F14	3.02*E*-11	3.02*E*-11	3.02*E*-11	3.02*E*-11	N/A	N/A	0.049941793	N/A	0.006518796

From the above on, regardless of the macroscopic or microscopic point of view, the time complexity has not changed, which undoubtedly proves the feasibility of the algorithm improvement.

5. Application in Constrained Engineering Optimization Problem

5.1. I-Shaped Beam. The design optimization problem of I-beam is one of the classic engineering optimization problems. The goal is to minimize the vertical deflection by optimizing the width of the leg x_1, the height of the waist x_2 and the two thicknesses (x_3, x_4). The objective function and constraint conditions of this optimization problem are as follows:

Minimize:

$$f(x) = \frac{5000}{x_3(x_2 - 2x_4)^3/12 + x_1x_4^3/6 + 2x_1x_4((x_2 - x_4)/2)^2}. \tag{18}$$

Subject to:

$$g_{(x)} = 2x_1x_3 + x_3(x_2 - 2x_4) \le 0. \tag{19}$$

Table 6: Best results for the optimal design of I-shaped beam problem.

Algorithm	Variables				Constraint		
	$x1$	$x2$	$x3$	$x4$	$g1(X)$	$g2(X)$	$f(X)$
IARSM	79.99	48.42	0.9	2.4	0.0869999	−1.52454	0.0131
CS	80	50	0.9	2.3216	−0.012005	−1.57002	0.01307
GWO	80	50	0.9	2.3217	−0.009059	−1.570071	0.0131
EMGO-FCR	80	50	0.9	2.32	−0.176	−1.567179	0.0131
SOS	80	50	0.9	2.3217	−0.000222	−1.570224	0.01307
AEFA-C	79.9671	49.99	0.9	2.3164	−0.560371	−1.559518	0.0131
SSA	79.99992	49.99982	0.9	2.321795732	−0.00058001	−1.570210836	0.013074174
IHSSA	80	50	0.9	2.32179226	−2.06E−08	−1.570228475	0.013074119

Table 7: Best results of the three-bar truss design problem.

Algorithm	Variables		Constraint			
	$x1$	$x2$	$g1(X)$	$g2(X)$	$g3(X)$	$f(X)$
GA	0.788915	0.407569	9.64E-07	−1.464873605	−0.53512542	263.8958857
PSO	0.788669	0.408265	4.8650E-07	−1.464082376	−0.535917137	263.8958434
ICA	0.788625	0.408389	8.42E-07	−1.463941244	−0.536057913	263.8958452
CS	0.78867	0.40902	−2.90E-04	−0.26853	−0.73176	263.9716
WCA	0.788651	0.408316	0.00E+00	−1.464024	−0.535975	263.895843
GWO	0.788648	0.408325	3.34E-08	−1.464014397	−0.535985569	263.8960063
ALO	0.788663	0.408283	−5.32E-12	−1.464062005	−0.53593799	263.8958434
MFO	0.788245	0.409467	7.71E-12	−1.462717072	−0.537282927	263.8959796
WSA	0.788683	0.408276	3.00E-10	−1.46407036	−0.53587454	263.8958434
SSA	0.788628	0.408381	5.43E-07	−1.463950108	−0.536049349	263.8957734
IHSSA	0.788674	0.408251	5.13E-09	−1.464098378	−0.535901617	263.8958427

Variable range:

$$\begin{aligned} 10 \le x_1 \le 50, \\ 10 \le x_2 \le 80, \\ 0.9 \le x_3 \le 5, \\ 0.9 \le x_4 \le 5. \end{aligned} \tag{20}$$

5.2. Tree-Bar Truss Design Problem. The design problem of three-bar truss is another classic problem in engineering case studies. In order to minimize the weight constrained by stress, deflection, and buckling, it is necessary to evaluate the optimal cross-sectional area and adjust the two long rods $A1$ and $A2$ (x_1, x_2). The specific mathematical formulas for adjustment are as follows:

Minimize:

$$f(x) = \left(2\sqrt{2}\, x_1 + x_2\right) \times l. \tag{21}$$

Subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + x_1 x_2} P - \sigma \le 0,$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + x_1 x_2} P - \sigma \le 0,$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \le 0,$$

$$\begin{aligned} l &= 100cm, \\ P &= \frac{2kN}{cm^3}, \\ \sigma &= \frac{2kN}{cm^3}. \end{aligned} \tag{22}$$

Variable range:

$$0 \le x_1, x_2 \le 1. \tag{23}$$

5.3. Cantilever Beam. The application is a structural engineering design problem. The component part of the cantilever arm is five hollow bricks, and the purpose of the project is to increase the rigidity. Increasing the cross-sectional height of the brickwork is more conducive to improving the rigidity. If the section height increases, in order to reduce the mass or maintain the same quality, the section width must be reduced. Therefore, the size of the cross section (height or width) is the optimal parameter for this experiment. The modeling expression of this case is as follows:

Minimize:

$$f(x) = 0.0624\left(x_1 + x_2 + x_3 + x_4 + x_5\right). \tag{24}$$

Table 8: Best results of the cantilever beam design example.

Algorithm	Variables					Constraint	
	$x1$	$x2$	$x3$	$x4$	$x5$	$g1(X)$	$f(X)$
CS	6.0089	5.3049	4.5023	3.5077	2.1504	−6.45E-05	1.33999
MFO	5.98487	5.31672	4.49733	3.51361	2.16162	4.18E-09	1.33998
ALO	6.01812	5.31142	4.48836	3.49751	2.15832	−3.00E-06	1.33995
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	1.39E-04	1.33996
SSA	5.99215	5.28536	4.54216468	3.482721286	2.174334383	−0.00014501	1.3401483
IHSSA	5.99349	5.33819	4.501471252	3.4892014	2.152033962	−1.92E-05	1.340002

Subject to:

$$g_{(x)} = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \le 0. \tag{25}$$

Variable range:

$$0.01 \le x_i \le 100, i = 1, \ldots, 5. \tag{26}$$

Three classic constrained engineering optimization problems, I-beam optimization problems, three-bar truss design problems, and cantilever beam problems, are representative in verifying the feasibility of the algorithm. The parameters and constraints of the three engineering problems are integrated in Tables 6–8, respectively. In decades of research [14, 35–41], to some extent, generations of researchers have designed many kinds of optimizers to solve these three nonlinear problems. The statistical results of these optimization methods (including the IHSSA proposed in this paper) are shown in Tables 6–8, respectively, and the optimal solutions obtained are denoted as $f(X)$. It can be seen from Tables 7 and 8 that the IHSSA algorithm can be used in engineering optimization problems and has better performance than the original SSA algorithm. Compared with other optimizers shown in [27], the overall result is also slightly superior.

6. Conclusion

Based on the basic sparrow search algorithm, this paper proposes an improved sparrow search algorithm (IHSSA) that integrates infinite folding iterative chaotic mapping and hybrid reverse learning strategy so as to deal with shortcomings. Firstly, an improved infinite fold iterative chaotic map (IICMIC) is introduced in the initial population stage to increase the search range of the population. Then, in order to update the position of the global optimal value and the current worst, a hybrid reverse learning strategy is proposed to be applied after the update of the discoverer and the update of the follower, respectively. The introduction of the hybrid reverse learning strategy increases the quality of understanding and avoids falling into the global optimum. Moreover, combining the vertical and horizontal crossover strategy into the monitor stage contributes to maximizing the exploration and mining capabilities of the balance algorithm. In general, the proposal of IHSSA makes the optimization accuracy better, the development ability becomes stronger, and the algorithm's global search ability gets enhanced.

Overall, the comparison results of the solutions obtained by the 14 standard test functions also prove that the new algorithm is generally better than several well-known heuristic algorithms such as WOA, GWO, TLBO, PSO, the newly proposed SSA, and its excellent variants. IHSSA has strong stability and robustness. In terms of running time, the optimization process of the seven functions takes the least amount of time, showing high computational efficiency. In addition, the high quality of convergence accuracy is proven in the Wilcoxon rank sum test. It is proven that the update of the algorithm does not bring an order of magnitude increase in time complexity, which indicates that it is a good operation. Moreover, the application of the improved algorithm in three constrained engineering optimization problems has demonstrated its great feasibility and effect, which is better than other optimizers. This undoubtedly makes the research more meaningful. However, IHSSA research is still in its infancy.

In the follow-up research, in order to obtain better accuracy and convergence speed, we will continue to try to improve the sparrow search algorithm and other swarm intelligence algorithms. In addition, the improved algorithm and innovative points are applied to engineering optimization problems to solve practical problems, so as to broaden the application field of the algorithm and further verify the feasibility and effectiveness of the algorithm.

Data Availability

Some data of our team needs to be kept confidential. If necessary, please ask the corresponding author for it.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by National key Research and Development program (Grant no. 2020YFB1713700).

References

[1] J. K. Xue, *Research and Application of a Novel Swarm Intelligence Optimization Technology*, DongHua University, Shanghai, China, 2020.

[2] X. Lv, X. D. Mu, J. Zhang, and Z. Wang, "Chaos sparrow search optimization algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 47, pp. 1–10, 2020.

[3] J. Wang, R. Hou, C. Wang, and L. Shen, "Improved v -Support vector regression model based on variable selection and brain

storm optimization for stock price forecasting," *Applied Soft Computing*, vol. 49, pp. 164–178, 2016.

[4] Z. Cai, S. Gao, X. Yang, Y. Gang, C. Shi, and S. Yuhui, "Alternate search pattern-based brain storm optimization," *Knowledge-Based Systems*, vol. 238, Article ID 107896, 2021.

[5] A. Khadwilard, S. Chansombat, T. Thepphakorn, P. Thapatsuwan, W. Thapatsuwan, and P. Pongcharoen, "Application of firefly algorithm and its parameter setting for job shop scheduling," *Journal of Industrial Technology*, vol. 8, no. 1, pp. 49–58, 2012.

[6] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[7] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of the ICNN'95-international conference on neural networks, vol. 4, pp. 1942–1948, IEEE, Perth, WA, Australia, December 1995.

[9] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the MHS'95 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, IEEE, Nagoya, Japan, October 1995.

[10] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[12] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.

[13] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the 2009 World congress on nature & biologically inspired computing (NaBIC)*, pp. 210–214, IEEE, Coimbatore, India, December 2009.

[14] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[15] Y. Zhu and N. Yousefi, "Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm," *International Journal of Hydrogen Energy*, vol. 46, no. 14, pp. 9541–9552, 2021.

[16] Q. Mao and Q. Zhang, "MixingSine and cosine algorithm with Levy flying chaotic sparrow algorithm," *Journal of Shanxi University (Nat.Sci.Ed.)*, vol. 1-6, 2020.

[17] G. Liu, C. Shu, Z. Liang, B. Peng, and L. Cheng, "A modified sparrow search algorithm with application in 3d route planning for UAV," *Sensors*, vol. 21, no. 4, p. 1224, 2021.

[18] J. Yuan, Z. Zhao, Y. Liu et al., "DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm," *IEEE Access*, vol. 9, pp. 16623–16629, 2021.

[19] T. Liu, Z. Yuan, L. Wu, and B. Benjamin, "Optimal brain tumor diagnosis based on deep learning and balanced sparrow search algorithm," *International Journal of Imaging Systems and Technology*, vol. 31, 2021.

[20] J. Zhang, K. Xia, Z. He, Z. Yin, and S. Wang, "Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6622935, 18 pages, 2021.

[21] C. Zhang and S. Ding, "A stochastic configuration network based on chaotic sparrow search algorithm," *Knowledge-Based Systems*, vol. 220, Article ID 106924, 2021.

[22] S. Zhou, H. Xie, C. Zhang et al., "Wavefront-shaping focusing based on a modified sparrow search algorithm," *Optik*, vol. 244, Article ID 167516, 2021.

[23] T.-Y. Li and J. A. Yorke, "Period three implies chaos," *The Theory of Chaotic Attractors*, Springer, New York, NY, pp. 77–84, 2004.

[24] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, 2019.

[25] Y. R. Wang and D. M. Zhang, "Butterfly optimization algorithm combining sine cosine and iterative chaotic map with infinite collapses," *Pattern Recognition and Artificial Intelligence*, vol. 33, no. 7, pp. 660–669, 2020.

[26] D. Di He, C. Chen He, L. G. Ling-Ge Jiang, Z. Hong-Wen, and H. Guang-Rui, "Chaotic characteristics of a one-dimensional iterative map with infinite collapses," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 7, pp. 900–906, 2001.

[27] A. A. Ewees, M. Abd Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Systems with Applications*, vol. 112, pp. 156–172, 2018.

[28] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Systems with Applications*, vol. 145, Article ID 113122, 2020.

[29] L. Sun, S. Chen, J. Xu, and T. Yun, "Improved monarch butterfly optimization algorithm based on opposition-based learning and random local perturbation," *Complexity*, vol. 2019, Article ID 4182148, 20 pages, 2019.

[30] D. M. Zhang, H. Xu, Y. R. Wang, T. Song, and L. Wang, "Whale optimization algorithm for embedded Circle mapping and onedimensional oppositional learning based small hole imaging," *Control and Decision*, vol. 36, no. 5, pp. 1173–1180, 2021.

[31] S. Shekhawat and A. Saxena, "Development and applications of an intelligent crow search algorithm based on opposition based learning," *ISA Transactions*, vol. 99, pp. 210–230, 2020.

[32] A.-b. Meng, Y.-c. Chen, H. Yin, and S.-z. Chen, "Crisscross optimization algorithm and its application," *Knowledge-Based Systems*, vol. 67, pp. 218–229, 2014.

[33] C. Ouyang, D. Zhu, and Y. Qiu, "Lens learning sparrow search algorithm," *Mathematical Problems in Engineering*, vol. 2021, no. 2, 17 pages, 2021.

[34] S. Yan, P. Yang, D. Zhu, W Zheng, and F Wu, "Improved sparrow search algorithm based on iterative local search," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 6860503, 31 pages, 2021.

[35] G. G. Wang, "Adaptive response surface method using inherited Latin hypercube design points," *Journal of Mechanical Design*, vol. 125, no. 2, pp. 210–220, 2003.

[36] I. Rechenberg, "Evolutionsstrategien, Simulationsmethoden in der Medizin und Biologie," *Workshop. (Sonderdruck Medizinische Informatik und Statistik*, Vol. 8, Springer, New York, NY, USA, 1978.

[37] A. Yadav and N. Kumar, "Artificial electric field algorithm for engineering optimization problems," *Expert Systems with Applications*, vol. 149, Article ID 113308, 2020.

[38] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: a new metaheuristic optimization algorithm," *Computers & Structures*, vol. 139, pp. 98–112, 2014.

[39] A. Kaveh and A. Dadras Eslamlou, "Water strider algorithm: a new metaheuristic and applications," *Structures*, vol. 25, pp. 520–541, 2020.

[40] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110-111, pp. 151–166, 2012.

[41] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 4719266, 12 pages
https://doi.org/10.1155/2022/4719266

Research Article

A Divide-and-Conquer Bat Algorithm with Direction of Mean Best Position for Optimization of Cutting Parameters in CNC Turnings

Xingwang Huang,[1] Zongbao He,[1] Yong Chen,[1] and Shutong Xie [1,2]

[1]*School of Computer Engineering, Jimei University, Xiamen 361021, China*
[2]*Digital Fujian Big Data Modeling and Intelligent Computing Institute, Jimei University, Xiamen 361021, China*

Correspondence should be addressed to Shutong Xie; shutong@jmu.edu.cn

Received 2 November 2021; Revised 4 January 2022; Accepted 13 January 2022; Published 23 February 2022

Academic Editor: Diego Oliva

Optimization of machining parameters is an important problem in the modern manufacturing world due to production efficiency and economics. This problem is well known to be complex and is regarded as a strongly nondeterministic polynomial (NP)-hard problem. To reduce the production cost of work-pieces in computer numerical control (CNC) machining, a novel optimization algorithm based on a combination of the bat algorithm and a divide-and-conquer strategy is proposed. First, the basic bat algorithm (BA) is modified with the aim to avoid finding the local optimal solution. In addition, a Gaussian quantum bat algorithm with direction of mean best position is developed. Second, in order to reduce the complexity of the optimization problem, the whole optimization problem is divided into several subproblems by using a divide-and-conquer strategy according to the characteristic of multipass turning operations. Finally, under a large number of machining constraints, the cutting parameters of the two stages of roughing and finishing are simultaneously optimized. Simulation results show that the proposed algorithm can find better combinations of the machining parameters than other algorithms proposed previously to further reduce the production cost. In addition, the outcome of our work presents a novel way to solve the complex optimization problem of machining parameters with a combination of traditional mathematical methods and swarm intelligence algorithms.

1. Introduction

In the manufacturing field, computer numerical control (CNC) machining refers to the computerized digital control of automated machine tools used to process rough material into semifinished or finished parts; it is one of the most common technologies. The main purpose of CNC machining is to save machining costs and improve machining efficiency and machining quality. Machining costs can be saved by selecting reasonable machining parameters, which introduces an optimization problem, i.e., selecting the optimal machining parameters to achieve the goal of reducing machining costs under the given machining constraints. Earlier research on the optimization of machining parameters mainly used traditional mathematical processing methods such as dynamic programming, sequential unconstrained minimization technique (SUMT), and linear or nonlinear programming. However, in general, the optimization problems of machining parameters are nonlinear and complicated problems with multiple constraints. Therefore, it is difficult to obtain satisfactory optimization solutions using traditional methods [1, 2]. In recent years, many scholars have applied swarm intelligence algorithms to the optimization problems of machining parameters in the field of computer integrated manufacturing. By using swarm intelligence algorithms to search for approximate optimal solutions of the problem, some research results have been achieved [3–16].

However, most previous studies were devoted to combining swarm intelligence algorithms with various local improvement algorithms [3–11, 17] (e.g., population diversification, local greedy search, and the use of heuristics as local search) in the hope of obtaining better results. However, because they did not fully consider the characteristics

of the turning problem with multiple machining processes, the results obtained by the algorithms were similar, and it was difficult to significantly reduce the machining cost. To address this bottleneck, this paper proposes a novel optimization algorithm by fully considering the characteristics of the turning problem while effectively exploiting the global optimization performance of the swarm intelligence algorithm. By combining the improved bat algorithm with the divide-and-conquer strategy, the performance of the optimization algorithm is substantially improved. The final optimization algorithm is able to find better results.

The rest of this paper is organized as follows. Section 2 represents the related works, especially the intelligent algorithms for optimization problems in CNC turning. Section 3 describes the mathematical model for the optimization of machining parameters in CNC turnings. Section 4 first introduces the bat algorithm, then proposes the Gaussian quantum bat algorithm with direction of mean best position (GQMBA), and finally elaborates on the idea of the combination of GQMBA and the divide-and-conquer strategy for solving the machining parameter optimization problems. In Section 5, simulation experiments are conducted, and different algorithms are compared. Finally, the concluding comments and some future research directions are presented in the last section.

2. Related Works

Optimization of turning parameters is an important issue in the manufacturing field. Early studies used traditional mathematical methods to find optimized machining parameters. Metaheuristic algorithms are also used to solve optimization problems of machining parameters. Chen and Tsai first proposed a mathematical model for the optimization problem of machining parameters in turnings and then combined Hooke-Jeeves pattern search (PS) into the simulated annealing (SA) algorithm to form a hybrid optimization algorithm (SA/PS) to solve the optimization problem [3]. Onwubolu and Kumalo [4] proposed a genetic algorithm (GA) to optimize the machining parameters in turnings but did not consider the constraint that the number of rough passes must be an integer. Chen and Chen [5] pointed out this shortcoming in the research of Onwubolu and Kumalo [4]. However, the optimization results obtained by the GA corrected by Chen and Chen were not better than those obtained by SA/PS. Additionally, based on a GA, Sankar [6] used a modified genetic algorithm (MGA) to search for optimized cutting parameters in turnings. The improved MGA used a specific crossover operator and three different mutation operators to enhance the diversity of the population and prevent the algorithm from converging to a local optimal solution. In addition to SA and GA, some studies applied other intelligent algorithms to the optimization problem of machining parameters. Vijayakumar [7], Wang [8], and Xie and Guo [16] developed new heuristic algorithms to overcome optimization problems based on the ant colony optimization (ACO) algorithm. In addition, the particle swarm optimization algorithm (PSO) is also one of the most widely used swarm intelligence methods [18]. Srinivas et al. proposed a PSO algorithm where the inertia coefficient decreased linearly with every iteration to solve the cutting parameter optimization problem [9]. Yildiz [10] and Costa et al. [12] also contributed different solutions based on the PSO algorithm to the problem. After comprehensive analysis of the previous research methods and results, Raja and Baskar [11] applied three optimization algorithms (SA, GA, and PSO) to three different machining parameter optimization models (single-pass turning, multipass turning, and surface grinding) to conduct experiments and compare the results of various types of intelligent optimization algorithms to the machining parameter optimization problem. The results showed that the optimization effect and computational efficiency of PSO are better than those of SA and GA. Scatter search (SS) is one of the optimization algorithms developed in the field of metaheuristics. Chen [19] focused on the application of the scatter search method in solving the optimization problem in turnings. By comparing it with other algorithms, the experimental results showed that the SS obtained superior machining parameters than some of the metaheuristic methods.

In recent years, in addition to the abovementioned classical intelligent algorithms, some new swarm intelligent algorithms have been proposed by researchers. Xu et al. proposed an improved flower pollination algorithm (FPA) and compared the obtained results with those of related studies [20]. Mellal and Williams used the cuckoo optimization algorithm (COA), one of the advanced bioinspired optimization algorithms, to minimize the unit production cost [21]. The experimental results showed that the COA algorithm is very competitive compared with other algorithms. Due to the successful application of swarm intelligence algorithms for optimization problems, Sofuoğlu et al. used three heuristic algorithms, GA, PSO and COA, to solve three different problems, which were more efficient and effective than other algorithms [22]. Similarly, Yildiz developed a new hybrid optimization algorithm to minimize the production cost by adding the Taguchi method that actively acted on the differential evolution algorithm to form a hybrid Taguchi-differential evolution algorithm (HRDE) [23]. The results showed that the hybrid algorithm was more effective than evolutionary algorithms presented in many related studies. In another work by Yildiz, he proposed a similar hybrid optimization method to determine the optimal machining parameters [24]. This method combined the differential evolution algorithm and receptor editing algorithm (DERE). The goal of the mathematical model was to determine the optimal machining parameters to reduce the unit production cost. The method has been experimentally proven to be an effective technique for optimizing machining parameters. Furthermore, in 2013, Yildiz proposed a parameter optimization method based on the artificial bee colony (ABC) algorithm [25] and a hybrid robust teaching-learning-based optimization algorithm (HRTLBO) based on the combination of guided learning optimization and the Taguchi method [26]. Compared with other methods, these proposed algorithms perform well, and better solutions can be found with them. Belloufi et al. provided specific application examples to illustrate the

effectiveness of the proposed firefly algorithm (FA) for parameter optimization in multipass turnings [27].

3. Mathematical Model for Optimizing Machining Parameters in CNC Turnings

To optimize the machining parameters in multipass turnings, the mathematical model proposed in the literature [3, 20] takes a large number of actual machining constraints into account and is closer to real-world machining. Since the model has been cited in many studies, the optimization model is used in this paper. The cutting parameters to be optimized include rough cutting speed V_r, rough feeding rate f_r, rough depth of cut d_r, the number of rough cuts n, finish cutting speed V_s, finish feeding rate f_s, and finish depth of cut d_s. The unit production cost (UC) consists of the following four components:

(1) Machining cost during real cutting time C_M

(2) Machine idle cost for setup operations and tool idling motion C_I

(3) Cost of tool replacement C_R

(4) Tool cost C_T

Thus, the UC can be expressed as follows:

$$\begin{aligned} UC &= C_M + C_I + C_R + C_T, \\ &= \left[\frac{\pi DL}{1000V_r f_r}\left(\frac{d_t - d_s}{d_r}\right) + \frac{\pi DL}{1000V_s f_s}\right]k_0, \\ &\quad + \left[t_c + (h_1 L + h_2)\left(\frac{d_t - d_s}{d_r} + 1\right)\right]k_0, \\ &\quad + \left[\frac{\pi DL}{1000V_r f_r}\left(\frac{d_t - d_s}{d_r}\right) + \frac{\pi DL}{1000V_s f_s}\right]\frac{t_e}{T_p}k_0, \\ &\quad + \left[\frac{\pi DL}{1000V_r f_r}\left(\frac{d_t - d_s}{d_r}\right) + \frac{\pi DL}{1000V_s f_s}\right]\frac{k_t}{T_p}. \end{aligned} \tag{1}$$

where k_0 is the sum of worker cost and management cost per unit time (\$/min). D and L are the diameter and length of the work-piece (mm), respectively. d_t is the depth of material to be removed (mm). h_1, h_2 are the constants related to tool idle time and tool-in/out time, respectively. t_c, t_e are the preparation time for loading and unloading time (min) and time required to exchange a tool (min), respectively. T_p is the tool life (min). k_t is the cutting edge cost (\$/edge).

The number of the rough cut is as follows:

$$n = \frac{d_t - d_s}{d_r},\ n \in z^+. \tag{2}$$

The objective of the model is to minimize the UC (V_r, V_s, f_r, f_s, d_r, d_s, n) under many machining constraints on rough and finish turnings. The constraints are summarized as follows [3–8]:

(1) The upper and lower constraints of V_r, V_s, f_r, f_s, d_r, and d_s

(2) Tool life constraints

(3) Cutting force, cutting power, and surface roughness constraint

(4) Stable cutting region constraint; chip-tool interface temperature constraint

(5) Constraints on the interconnection between roughing and finishing parameters

4. Optimization Algorithm Based on the Bat Algorithm and the Divide-and-Conquer Strategy

4.1. Overview of the Bat Algorithm. The bat algorithm (BA) [28] is a swarm intelligence search algorithm proposed to simulate the echolocation mechanism of bats when foraging. It achieves the localization of search targets by continuously adjusting the frequency and loudness of sound waves. It uses frequency tuning to increase the population diversity and uses automatic scaling to maintain a balance between global and local searches. The frequency, velocity, and position of the i-th bat in the bat population are represented as follows:

$$\begin{aligned} f_i &= f_{\min} + (f_{\max} - f_{\min})\beta, \\ v_i^t &= v_i^{t-1} + \left(v_i^t - gb^t\right)f_i, \\ x_i^t &= x_i^{t-1} + v_i^t. \end{aligned} \tag{3}$$

Here, f_i denotes the ultrasound frequency emitted by the i-th bat. $f_{\max}$ and $f_{\min}$ denote the upper and lower bounds of ultrasound frequency, respectively. β denotes a random number generated by a uniform distribution within the range of [0,1]. v_t^t and v_t^{t-1} denote the velocity of the i-th bat during the t-th and (t−1)-th iterations, respectively. x_i^t and x_i^{t-1} denote the position value of the i-th bat during the t-th and (t−1)-th iterations, respectively. The above expressions ensure the global search capability of the algorithm.

In the local search phase, BA uses a random walk strategy to generate feasible solutions at candidate locations. This strategy can be given by the following equation:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon \overline{A}^t. \tag{4}$$

where ε is the random number generated by the uniform distribution on the range [−1, 1] that determines the direction of the new candidate feasible solution and $\overline{A}^t$ denotes the average acoustic loudness of all bats in the t-th iteration.

During the foraging process, as the bat approaches the foraging target, the bat will gradually adjust the loudness A and emission rate r of the ultrasound, making the loudness gradually decrease and the emission rate gradually increase to achieve more accurate positioning. The process is shown in the following equations:

$$A_i^t = \alpha A_i^{t-1}. \tag{5}$$

$$r_i^t = r_i^0\left[1 - \exp(-\gamma t)\right]. \tag{6}$$

In formula (6), r_i^0 denotes the initial pulse emission rate of the i-th bat, and both α and γ are constants between (0, 1).

4.2. Overview of the Quantum-Behaved Bat Algorithm with Mean Best Position Directed. Due to the lack of population diversity in the original BA, there is the problem of falling into local optima during the search. By analyzing the flight trajectory of bats, Zhu et al. proposed the quantum-behaved bat algorithm with mean best position directed (QMBA) [29]. The quantum computing mutation operator introduced by the algorithm can enhance population diversity and avoid premature convergence. At the same time, its average optimal position introduced in the local search phase can improve the convergence speed in the later stage of the search. The QMBA still retains the main body of the BA, which controls the global search and local search based on the ultrasonic loudness and the sending rate. The difference lies in its improved position update formula and local search strategy.

The position update formula in QMBA introduces a mechanism for adaptively adjusting the step size according to the distance. Its strategy for updating the position is expressed as follows:

$$x_{i\,d}^{t} = \begin{cases} x_{i\,d}^{t-1} + \left(gb_d - x_{i\,d}^{t-1}\right)\eta, & \delta_d > TH \\ x_{i\,d}^{t-1} + \epsilon, & \delta_d \le TH \end{cases}, \tag{7}$$

where η is the random number generated through the uniform distribution probability function between [0,1]. δ_d represents the distance between the d-th dimensional value of the current global optimum position and the d-th dimensional value of the i-th bat, which is mathematically expressed as follows:

$$\delta_d = \left|gb_d - x_{i\,d}^{t-1}\right|. \tag{8}$$

When δ_d is less than a given threshold TH, the i-th bat can fly for food at will; when the value of δ_d is greater than a given threshold TH, the i-th bat flies toward the current global optimal position. This strategy ensures the global search ability of the bat population to fly toward food.

In the local search process, QMBA no longer uses the random walk search strategy but decides the selection of the mutation strategy based on the mutation probability p_{m}.

The first mutation strategy is to use the quantum computing mutation operator, which is expressed as follows:

$$x_{i\,d}^{t} = \begin{cases} gb_d^t + \mu \times \left|m\mathrm{best}_d - x_{i\,d}^{t}\right| \times \ln(1/U), r\mathrm{and} < 0.5 \\ gb_d^t - \mu \times \left|m\mathrm{best}_d - x_{i\,d}^{t}\right| \times \ln(1/U), r\mathrm{and} \ge 0.5 \end{cases}, \tag{9}$$

where both U and *rand* are the random numbers generated by the uniform distribution probability function between [0,1]. μ is an adaptive linearly decreasing weighting factor that can be expressed as follows:

$$\mu = \mu_{\max} - \frac{t\left(\mu_{\max} - \mu_{\min}\right)}{t_{\max}}. \tag{10}$$

where $\mu_{\max}$ and $\mu_{\min}$ denote the initial and final values of μ, respectively.

*m*best represents the average of the current optimal position of all bats during the t-th iteration, i.e., the average optimal position, which can be obtained from the following equation:

$$\mathrm{mbest} = \frac{1}{M}\left(\sum_{i=1}^{M} P_{i1}^{t}, \sum_{i=1}^{M} P_{i2}^{t}, \cdots, \sum_{i=1}^{M} P_{i\,D}^{t}\right), \tag{11}$$

where P_i^t denotes the current optimal position of the i-th bat, M denotes the population size, and D denotes the dimension of the problem.

The second mutation strategy, which also introduces the average optimal position into the mutation operator, is expressed as follows:

$$x_i^t = x_i^{t-1} + \left(\mathrm{mbest} - x_i^{t-1}\right)\phi, \tag{12}$$

where ϕ denotes the random number generated by a uniform distribution between [0,1].

Both mutation strategies introduce the average optimal position to guide the local search, which can improve the accuracy of the search and speed up the convergence of the algorithm due to the use of statistical information of bat positions.

4.3. Gaussian Quantum Bat Algorithm with Direction of Mean Best Position. The QMBA algorithm improves the global search capability and accuracy of the algorithm by introducing the mechanism of the distance adaptive adjustment step, quantum computing mutation operator, and average optimal position-oriented mechanism on the basis of BA. However, the probability density functions used to generate the random numbers in QMBA are all uniformly distributed. Several works [30, 31] have shown that long-tailed distributions such as Gaussian distributions are able to perform more accurate searches in the region near the previous generation of individuals, improving the local search capability while providing larger search steps and random walk distances. Expanding the search space can improve the ability of the algorithm to jump out of the local optimum. Based on the above findings, this paper proposes the Gaussian quantum bat algorithm with direction of mean best position (GQMBA) for quantum behavior bats using a Gaussian distribution [32].

In GQMBA, random numbers are no longer generated by the uniformly distributed probability density function. To meet the requirements of the quantum computing mutation operator for random numbers in QMBA, we use the absolute value of the Gaussian distribution probability density function, in which the mean is zero and the variance is one instead (i.e., normal distribution). The one-dimensional probability density function of abs(N(0, 1)) is expressed as follows:

$$q(x) = \frac{2}{\sqrt{2\pi}} exp\left(-\frac{x^2}{2}\right), \; x \ge 0, \tag{13}$$

GQMBA modifies the three formulas in QMBA accordingly. First, the parameter η in (14) is changed to be generated with a Gaussian distribution, and (7) is modified in GQMBA as follows:

$$x_{id}^{t} = \begin{cases} x_{id}^{t-1} + \left(gb_d - x_{id}^{t-1}\right)G, & \delta_d > TH \\ x_{id}^{t-1} + \epsilon, & \delta_d \le TH \end{cases}, \tag{14}$$

where $G = abs(N(0,1))$.

Similarly, substituting for the parameter U in (9), the modified quantum computing mutation operator is expressed as follows:

$$x_{id}^{t} = \begin{cases} gb_d^t + \mu \times \left|m\text{best}_d - x_{id}^t\right| \times \ln(1/G), & r\text{and} < 0.5 \\ gb_d^t - \mu \times \left|m\text{best}_d - x_{id}^t\right| \times \ln(1/G), & r\text{and} \ge 0.5 \end{cases} \tag{15}$$

Since $q(0) = 0$, $G = abs(N(0,1))$ satisfies the domain of definition of the function ln().

Finally, the random number ϕ in (12) is replaced with (16) as follows:

$$x_i^t = x_i^{t-1} + \left(\text{mbest} - x_i^{t-1}\right)G. \tag{16}$$

The pseudocode of GQMBA is given by Figure 1, in which Np denotes the total number of bats.

Our first significant contribution is that the Gaussian distribution is introduced in QMBA to generate random numbers. The theoretical analysis above shows that the strategy can enhance the ability of the algorithm to jump out of the local optimum and avoid premature convergence. Therefore, it is applied to the optimization problem of the cutting parameter in this paper.

4.4. Divide-and-Conquer Strategy for the Optimization Problem in Multipass Turnings. To improve the performance of the algorithm, the idea of the divide-and-conquer strategy is used to decompose the original problem into several subproblems, which can reduce the complexity of the original optimization problem. For each subproblem, the number of rough cuts is a fixed value. By conquering the subproblems one by one, the whole optimization problem can be solved. In addition, we calculate the theoretical lower bound on UC for each subproblem. Modified BA is first used to search for the optimal solution in the case of the minimum theoretical lower bound on UC, thus hopefully reducing the enumeration of the subproblems. The divide-and-conquer strategy is depicted in Figure 2 and described as follows:

(1) Divide the optimization problem into m subproblems based on the number of possible combinations of rough cuts.

(2) Calculate the theoretical lower bound on UC for each subproblem UC_{iL}.

(3) Sort theoretical lower bound UC_{iL} for all subproblems in ascending order. $UC_{1L} \le UC_{2L}, \ldots, \le UC_{mL}$ are called the first theoretical lower bound, the second theoretical lower bound ..., the m-th theoretical lower bound, and the corresponding numbers of rough cuts $N_1, N_2, \ldots, N_m$ (N_i is the number of rough cuts corresponding to UC_{iL}) [16].

(4) Starting from subproblem i, BA is used to solve subproblem i, and the optimal solution, UC_{iO}, is obtained.

(5) If all subproblems are enumerated or the UC_{iO} found is less than the theoretical lower bound of subsequent subproblems $UC_{(i+1)\mathrm{L}}$, the method terminates and the optimal solution is output.

4.5. The Framework of the Proposed Algorithm Based on GQMBA and the Divide-and-Conquer Strategy. By dividing the complicated multipass turning optimization problem into simple subproblems, the optimization problem can be solved by solving these subproblems one by one. The framework of the optimization algorithm based on GQMBA and the divide-and-conquer strategy (referred to as the GQMBA-DC algorithm) is shown in Figure 3, and the main steps are as follows:

(1) Divide the optimization problem into n subproblems based on the number of possible combinations of rough cuts.

(2) Let $i = i + 1$; set the number of rough cuts $n = N_i$; and start the search in the i-th subproblem.

(3) Initialize the population, develop appropriate encoding and decoding strategies for each subproblem for GQMBA, and set the current iteration number $t = 1$.

(4) Initialize the parameters and set the ultrasonic frequency f_i, ultrasonic emission rate r_i, and ultrasonic loudness A_i.

(5) The global search and local search are controlled by continuously adjusting the acoustic frequency and loudness to update the speed and position to generate new solutions. For the GQMBA-DC, the main body of the BA is retained, but the position update formula and the local search strategy are different. The position update formula introduces a mechanism for adaptively adjusting the step size according to the distance, while the local search strategy also introduces the average optimal position to guide the local search.

(6) If A_i is greater than the random value *rand* and the current solution is the optimal solution, perform the next step; otherwise, return to step (5).

(7) Accepting the new solution increases r_i and decreases A_i (as a bat gets closer to the target, the two values change to achieve more accurate localization).

(8) Repeat the above steps until the maximum number of iterations is reached, and the GQMBA stops.

(9) At present, the optimal solution of the i-th subproblem is obtained.

(10) If $Min(UC_{1O}..., UC_{iO}) \le U_{(i+1)L}$ or $i = m$, then execute the next step. Otherwise, return to step (2) to start the process of solving the next subproblem.

Algorithm 1: Pseudocode of GQMBA.

Initialize the position x_i and velocity v_i of bats ($i = 1,2,\cdots,Np$);

Setup the pulse frequency f_i, pulse rate r_i and the loudness A_i;

while ($t < t_{max}$) **do**

for $i = 1\ to\ Np$ **do**

Calculate the distance between the position of i-th bat (x_i) and the current global best position, and then update the position by using formula (10) and formula (16);

if (rand $>$ r_i) **then**

if (rand $>$ p_m) **then**

This bat flies with quantum behavior and its position is updated using formulas (12), (13) and (17);

else

Guided by the mean best position, the position of i-th bat is updated using formulas (13) and (18);

end if

end if

if (rand$<$ A_i && $f(x_i) < f(gb)$) **then**

Accept the new solution to update x_i;

Increate r_i and reduce A_i using formulas (5) and (7), respectively;

end if

Rank bats to find the current global best location gb;

end for

$t = t + 1$;

end while

Figure 1: Pseudocode of GQMBA.

(11) Select the minimum solution, which is the best optimal solution, from the obtained optimized solutions. Finally, output the global optimal solution $UC_o = Min(UC_{1O}, \ldots, UC_{iO})$, and terminate the algorithm.

4.6. Handling of Constraints. The processing of constraints is very important for the swarm intelligence optimization algorithm; constraint processing by adding a penalty function is one of the common methods in optimization algorithms. The penalty function is a kind of constraint function. In the process of finding the optimal solution of the algorithm, the objective function is calculated by combining the penalty function, which can gradually eliminate solutions that do not satisfy the constraints and retain solutions that satisfy the constraints.

For the handling of constraints in the optimization algorithm, the bats (individuals) that violate the constraints are penalized using a penalty function to reduce the value of the objective. Different levels of penalties are imposed for different constraint violations. The more constraints that are violated, the heavier the penalty will be. Thus, by using a reasonable penalty function, the objective function value can converge to the direction of the optimal solution. The penalty function is expressed as

$$\text{penalty}(X) = \sum_{i=1}^{k} a_i + h_i,\ \text{where}\ a_i = \begin{cases} 0, & \text{satisfy constraints} \\ 1, & \text{violate constraints} \end{cases}, \tag{17}$$

where k is the number of constraints and h_i is a nondimensional constraint violation.

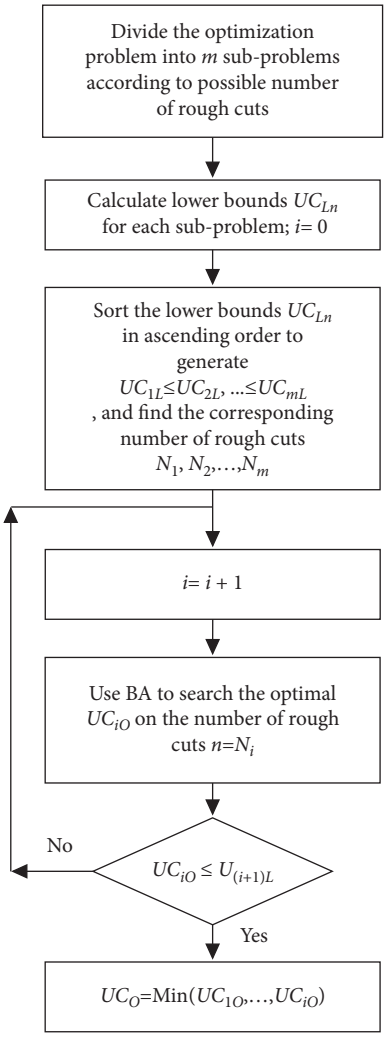


Figure 2: Flow chart of the divide-and-conquer strategy.

5. Simulation Experiments

During the machining processes, a cut tool is used for both roughing cuts and finishing cuts. Due to different machining conditions, the tool wear rates for rough and finish turnings are usually different. The tool life equation can be expressed as follows: $T_p = \theta T_r + (1 - \theta) T_s$. Some studies use another tool life calculation formula: $T_p = T_r + T_s$. Therefore, this paper separately compares the performance of the algorithms under different tool life formulations. The algorithm in this paper is implemented in the MATLAB programming language. The parameters of the BA are set as follows:

Population size: 200

Maximum number of iterations: 400

Initial loudness: $A = u(0, 1)$

Initial pulse emission rate: $r_0 = 0.001$

Loudness update: $\alpha = 0.9$

Emission rate update: $\gamma = 0.9$

Threshold: $TH = 0.005$

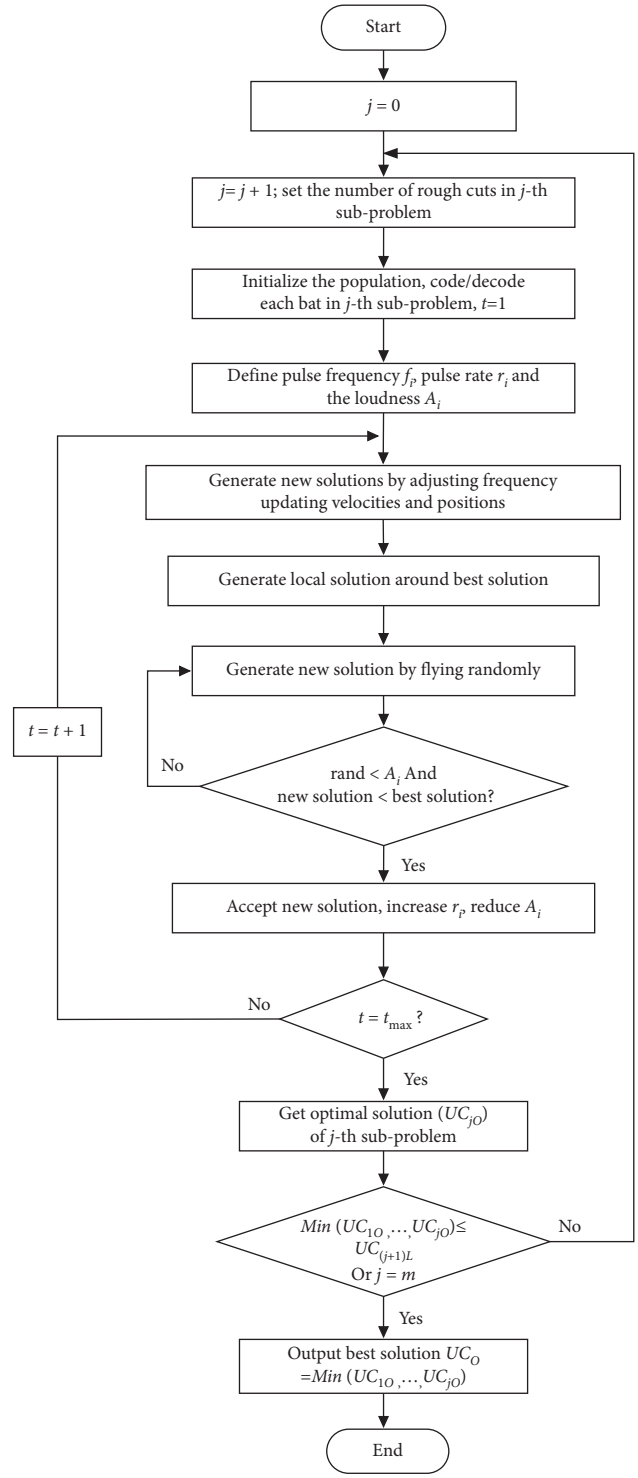


Figure 3: The framework of the proposed algorithm (GQMBA-DC) based on GQMBA with a divide-and-conquer strategy.

Mutation probability: $P_m = 0.01$

Machining examples from the literature [3–8] were used to test the performance of the optimization algorithm with the specific parameters shown in Table 1. Additionally, two different machining optimization problems with cutting depths of 6 mm and 8 mm were tested.

Table 1: Condition parameters for turning examples.

$D = 50$ mm	$L = 300$ mm	$d_t = 6$ mm	$V_{rL} = 50$ m/min
$f_{rL} = 0.1$ mm/rev	$d_{rL} = 1$ mm	$V_{rU} = 500$ m/min	$f_{rU} = 0.9$ mm/rev
$d_{rU} = 3$ mm	$V_{sL} = 50$ m/min	$f_{sL} = 0.1$ mm/rev	$d_{sL} = 1$ mm
$V_{sU} = 500$ m/min	$f_{sU} = 0.9$ mm/rev	$d_{sU} = 3$ mm	$p = 5$
$q = 1.75$	$r = 0.75$	$\mu = 0.75$	$\nu = 0.95$
$\eta = 0.85$	$\lambda = 2$	$\upsilon = -1$	$\tau = 0.4$
$\phi = 0.2$	$\delta = 0.105$	$R = 1.2$ mm	$C_0 = 6\times1011$
$T_L = 25$ min	$T_U = 45$ min	$F_U = 200$ kgf	$P_U = 5$ kW
$SC = 140$	$Q_U = 1000$ °C	$SR_U = 10$ μm	$h_1 = 7\times10-4$
$h_2 = 0.3$	$t_e = 1.5$ min/edge	$t_c = 0.75$ min/piece	$k_t = 2.5$ \$/edge
$k_0 = 0.5$ \$/min	$k_1 = 108$	$k_2 = 132$	$k_3 = 1$
$k_4 = 2.5$	$k_5 = 1$		

The algorithm was run 100 times independently on a Windows platform (CPU E3 3.5 GHz and 16 GB memory). The average value of UC was given and compared with the results obtained by previous algorithms, such as SA/PS [3], FE-GA [5], MGA [6], ACO [6], and PSO [9]. The average UC, standard deviation, number of search points, and running time for each algorithm are shown in Tables 2-5. The best results have been underlined and bolded in Tables 2-5.

Tables 2–4 show that the average UCs obtained by the proposed GQMBA-DC are smaller than those given by other algorithms. The standard deviations of the results are small, which in turn indicates that the algorithm is stable. The proposed algorithm can find the optimization results within 30 seconds for different tool life formulas and cutting depths, which shows that the proposed algorithm is an efficient algorithm. Specifically, as shown in Tables 2 and 3, the results of both cases of GQMBA-DC outperform PSO [9] when the tool life equation is $T_p = T_r + T_s$. As shown in Table 4, when the tool life equation is $T_p = \theta T_r + (1-\theta)T_s$, the proposed GQMBA-DC can save 10% compared with the result given by the MGA [6]. Compared with other algorithms, such as SA/PS [3], FE-GA [5], HC [6], NM [6], ACO [6], and DP-FS [33], GQMBA-DC can further save production costs. Because the case of $d_t = 8$ cm is not covered in previous literature, only the results of our algorithm are given in Table 5. Thus, the above experimental results show that the GQMBA-DC algorithm can effectively solve the optimization problem of cutting parameters to find optimal machining parameters, which, in turn, can further reduce the production cost.

From the perspective of the optimal UC value, comparisons between the proposed GQMBA-DC and other algorithms were also conducted. The comparison of the optimal UC values in Tables 6–9 shows that the optimal UC results obtained by the GQMBA-DC are almost always smaller than the optimal results of UC obtained by other algorithms without constraint violation. The best results have been underlined and bolded in Tables 6–9. Specifically, the results found by the proposed algorithm are comparable to those achieved by HPSO [12], FPA [20], and COA [21] for the tool life equation of $T_p = T_r + T_s$ and $d_t = 6$ mm, which is only one ten-thousandth of the difference, as shown in Table 6. The GQMBA-DC can further reduce the production cost compared with the other algorithms (i.e., HRDE [23], DERE [24], DE [24], and HRTLBO [26]), as shown in Table 6. A similar situation can also be found in Tables 7–9 in different test examples. In addition, the optimal combination of cutting parameters (V_r, V_s, f_r, f_s, d_r, d_s) for different cases is also given in Tables 6–9. The GQMBA-DC algorithm can find better results than the previously proposed algorithms in terms of both the average UC and the best UC. Thus, it is clear that the proposed GQMBA-DC can perform significantly better than other algorithms on solution quality in CNC turnings. Therefore, the algorithm combining the modified BA with the divide-and-conquer strategy is effective.

Table 2: Comparison of average UC among different algorithms (when $T_p = T_r + T_s$, $d_t = 6$ mm).

Algorithm	Average UC (\$)	Standard deviation	Search points (pcs)	Running time (sec)
PSO [9]	>2.2721	N/A	2,000	N/A
GQMBA-DC	**1.9592**	0.00005	80,000	28.4

Table 3: Comparison of average UC among different algorithms (when $T_p = T_r + T_s$, $d_t = 8$ mm).

Algorithm	Average UC (\$)	Standard deviation	Search points (pcs)	Running time (sec)
PSO [9]	>3.306	N/A	2,000	N/A
GQMBA-DC	**2.4393**	0.00084	80,000	30.4

To overcome the different complex optimization problems in various fields, we need to carefully consider the characteristics of the specific problem and use the specific characteristics (domain knowledge) to design the optimization algorithm. In our work, for the optimization problem of machining parameters, since the machining process can be divided into different numbers of roughing cuts, we decompose the whole optimization problem of machining parameters into several simple subproblems according to the different numbers of roughing cuts. Each subproblem can be conquered individually, which greatly reduces the space of the problem solution. At the same time, to avoid enumerating all subproblems and save calculation time, we derived the theoretical lower bound on UC for each subproblem by

Table 4: Comparison of average UC among different algorithms (when $T_p = \theta T_r + (1\text{-}\theta)T_s$, $d_t = 6$ mm).

Algorithm	Average UC ($)	Standard deviation	Search points (pcs)	Running time (sec)
SA/PS [3]	2.2959	0.01624	$18{,}571 \times 5$	27.4
FE-GA [5]	2.3091	N/A	60,000	N/A
HC [6]	2.3017	N/A	100,000	N/A
NM [6]	2.2713	N/A	100,000	N/A
ACO [6]	2.2705	N/A	100,000	N/A
MGA [6]	2.2538	N/A	100,000	N/A
DP-FS [33]	2.2974	7.6×10^{-4}	16074×9	19.3
GQMBA-DC	**2.0280**	0.00015	80,000	27.9

Table 5: Comparison of average UC among different algorithms (when $T_p = \theta T_r + (1\text{-}\theta)T_s$, $d_t = 8$ mm).

Algorithm	Average UC ($)	Standard deviation	Search points (pcs)	Running time (sec)
GQMBA-DC	**2.5499**	0.00043	80,000	29.2

Table 6: Comparison of different algorithms (when $T_p = T_r + T_s$, $d_t = 6$ mm).

Algorithm	Cutting speed (m/min)		Feed rate (mm/rev)		Depth of cut (mm)		UC ($/piece)	Constraint violation
	V_r	V_s	F_r	f_s	d_r	d_s		
GQMBA-DC	123.3360	169.9697	0.5655	0.2262	3	3	**1.9592**	0
HPSO [12]	123.3424	169.9783	0.5655	0.2262	3	3	**1.9591**	0
FPA [20]	123.3431	169.9785	0.5655	0.2262	3	3	**1.9591**	0
COA [21]	123.1462	169.9876	0.5655	0.2262	3	3	**1.959**	0
GA [4]	1114.22	164.369	0.7	0.2978	2.9745	2.9863	1.7842	0.5148
ACO [7]	103.05	162.02	0.9	0.24	3	3	1.8450	0.5396
PSO [9]	106.69	155.89	0.897	0.28	2	2	2.2721	0
HRDE [23]	–	–	–	–	–	–	2.0461	–
AIA [23]	–	–	–	–	–	–	2.12	–
DERE [24]	–	–	–	–	–	–	2.046	–
ABC [24]	–	–	–	–	–	–	2.118	–
DE [24]	–	–	–	–	–	–	2.136	–
HABC [25]	–	–	–	–	–	–	2.046	–
HRTLBO [26]	–	–	–	–	–	–	2.046	–
FA [27]	98.4102	162.2882	0.82	0.2582	3	3	1.824	(24)

Table 7: Comparison of different algorithms (when $T_p = T_r + T_s$, $d_t = 8$ mm).

Algorithm	Cutting speed (m/min)		Feed rate (mm/rev)		Depth of cut (mm)		UC ($/piece)	Constraint violation
	V_r	V_s	f_r	f_s	d_r	d_s		
GQMBA-DC	119.1460	164.2166	0.6564	0.2625	2.6673	2.6613	**2.4384**	0
HRDE [23]	–	–	–	–	–	–	2.4791	–
AIA [23]	–	–	–	–	–	–	2.51	–
DERE [24]	–	–	–	–	–	–	2.4793	–
HABC [25]	–	–	–	–	–	–	2.4790	–
ABC [24]	–	–	–	–	–	–	2.503	–
DE [24]	–	–	–	–	–	–	2.512	–

using the characteristics of the subproblems. Then, the algorithm first searches the solution space from the subproblem with a smaller theoretical lower bound on UC. By following these steps, the algorithm can quickly find the optimal solution to the problem.

On the other hand, the performance of the combination of traditional divide-and-conquer strategy and swarm intelligence algorithm is better than the algorithms that only use traditional mathematical methods or swarm intelligence algorithms, as proven by the simulation experiments.

Table 8: Comparison of different algorithms (when $T_p = \theta T_r + (1\text{-}\theta)T_s$, $d_t = 6$ mm).

Algorithm	Cutting speed (m/min)		Feed rate (mm/rev)		Depth of cut (mm)		UC ($/piece)	Constraint violation
	V_r	V_s	f_r	f_s	d_r	d_s		
GQMBA-DC	109.6727	169.9756	0.5655	0.2262	3	3	**2.0278**	0
SA-PS [3]	–	–	–	–	–	–	2.3135	0.0667
HPSO [9]	109.6655	169.9796	0.5655	0.2262	3	3	2.0351	0
FPA [20]	109.6631	169.9785	0.5655	0.2262	3	3	2.0351	0
COA [21]	117.9322	123.1993	0.5655	0.2262	3	3	2.2390	0

Table 9: Comparison of different algorithms (when $T_p = \theta T_r + (1\text{-}\theta)T_s$, $d_t = 8$ mm).

Algorithm	Cutting speed (m/min)		Feed rate (mm/rev)		Depth of cut (mm)		UC ($/piece)	Constraint violation
	Vr	Vs	Fr	fs	Dr	ds		
GQMBA-DC	106.0251	164.2238	0.6563	0.2624	2.6670	2.6660	**2.5495**	0
SA-PS [3]	–	–	–	–	–	–	2.7411	0

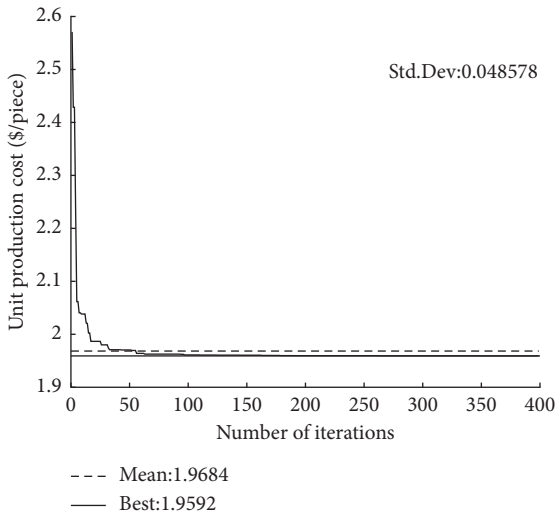


Figure 4: Convergence curve of the proposed GQMBA-DC (when $T_p = T_r + T_s$, $d_t = 6$ mm).

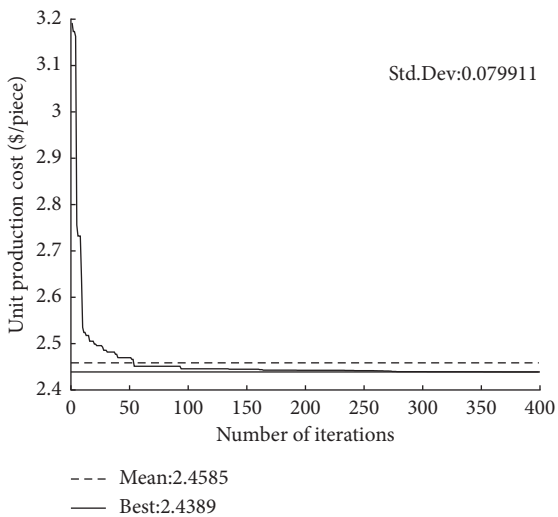


Figure 5: Convergence curve of the proposed GQMBA-DC ($T_p = T_r + T_s$, $d_t = 8$ mm).

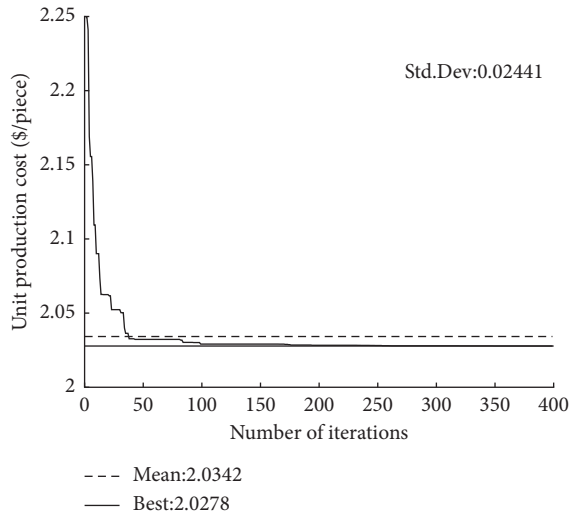


Figure 6: Convergence curve of the proposed GQMBA-DC (when $T_p = \theta T_r + (1\text{-}\theta)T_s$, $d_t = 6$ mm).

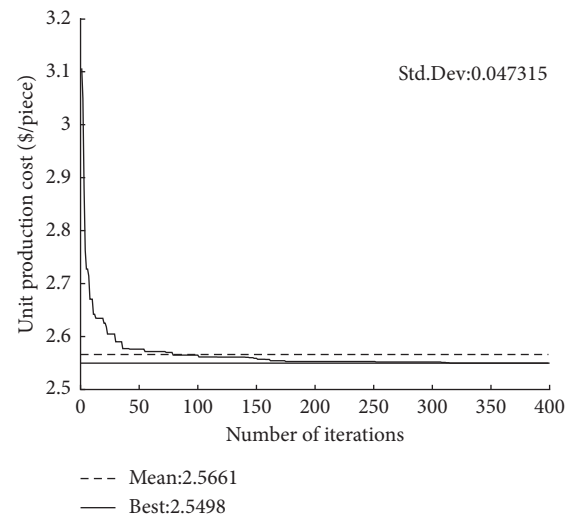


Figure 7: Convergence curve of the proposed GQMBA-DC (when $T_p = \theta T_r + (1\text{-}\theta)T_s$, $d_t = 8$ mm).

The convergence curves of GQMBA-DC for different mathematical models (tool life equation) and test cases are shown in Figures 4–7. The proposed algorithm converges to the final solution after approximately 150 generations, which indicates that the algorithm converges quickly to find satisfactory results.

6. Conclusions and Future Work

To solve the nonlinear optimization problem of machining parameters in CNC turnings, this paper proposes an optimization algorithm combining the bat algorithm and the divide-and-conquer strategy. First, based on the classical BA, the Gaussian quantum bat algorithm with direction of mean best position (GQMBA) is proposed by using a Gaussian distribution to generate random numbers. Second, the divide-and-conquer strategy is used to divide the complicated optimization problem into several subproblems and conquer them one by one. The simulation results show that the GQMBA-DC algorithm proposed in this paper has a stronger search capability than previous algorithms. Specifically, the proposed algorithm can find a better cutting parameter set and further reduce the production cost.

Future research can be considered from two aspects. From the algorithmic point of view, the emerging swarm intelligence algorithm can also be applied to the optimization problem, which may be able to find a better combination of machining parameters, thus reducing costs. In recent years, deep learning methods have been widely applied in various studies; deep learning methods may be considered to reconstruct mathematical models in the optimization of turning parameters [34]. On the other hand, from the perspective of new machining types, to improve the machining efficiency and quality, there are multiple tools to realize machining operations simultaneously in modern CNC turnings. Therefore, research on this type of machining optimization problem is also of great concern.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Fujian Province of China (Nos. 2020J01699 and 2020J01697).

References

[1] M. Chandrasekaran, "Application of soft computing techniques in machining performance prediction and optimization: a literature review," *International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5, pp. 445–464, 2010.

[2] I. Mukherjee and P. K. Ray, "A review of optimization techniques in metal cutting processes," *Computers & Industrial Engineering*, vol. 50, no. 1-2, pp. 15–34, 2006.

[3] M.-C. Chen and D.-M. Tsai, "A simulated annealing approach for optimization of multi-pass turning operations," *International Journal of Production Research*, vol. 34, no. 10, pp. 2803–2825, 1996.

[4] G. C. Onwubolu and T. Kumalo, "Optimization of multipass turning operations with genetic algorithms," *International Journal of Production Research*, vol. 39, no. 16, pp. 3727–3745, 2001.

[5] M.-C. Chen and K.-Y. Chen, "Optimization of multipass turning operations with genetic algorithms: a note," *International Journal of Production Research*, vol. 41, no. 14, pp. 3385–3388, 2003.

[6] R. S. Sankar, "Selection of machining parameters for constrained machining problem using evolutionary computation," *International Journal of Advanced Manufacturing Technology*, vol. 32, no. 9-10, pp. 892–901, 2007.

[7] K. Vijayakumar, "Optimization of multi-pass turning operations using ant colony system," *International Journal of Machine Tools and Manufacture*, vol. 43, no. 15, pp. 1633–1639, 2003.

[8] Y.-C. Wang, "A note on optimization of multi-pass turning operations using ant colony system," *International Journal of Machine Tools and Manufacture*, vol. 47, no. 12-13, pp. 2057–2059, 2007.

[9] J. Srinivas, R. Giri, and S.-H. Yang, "Optimization of multi-pass turning using particle swarm intelligence," *International Journal of Advanced Manufacturing Technology*, vol. 40, no. 1, pp. 56–66, 2009.

[10] A. R. Yıldız, "A novel particle swarm optimization approach for product design and manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 40, no. 5, pp. 617–628, 2009.

[11] S. B. Raja and N. Baskar, "Optimization techniques for machining operations: a retrospective research based on various mathematical models," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9, pp. 1075–1090, 2010.

[12] A. Costa, G. Celano, and S. Fichera, "Optimization of multi-pass turning economies through a hybrid particle swarm optimization technique," *International Journal of Advanced Manufacturing Technology*, vol. 53, no. 5, pp. 421–433, 2011.

[13] A. Jeang, H.-C. Li, and Y.-C. Wang, "A computational simulation approach for optimising process parameters in cutting operations," *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 4, pp. 325–340, 2010.

[14] G. Singh, "An evolutionary approach for multi-pass turning operations," *Proceedings of the Institution of Mechanical Engineers - Part B: Journal of Engineering Manufacture*, vol. 220, no. 2, pp. 145–162, 2006.

[15] L. Tang, R. G. Landers, and S. N. Balakrishnan, "Parallel turning process parameter optimization based on a novel heuristic approach," *Journal of Manufacturing Science and Engineering-Transactions of the Asme*, vol. 130, no. 3, pp. 031002–0310012, 2008.

[16] S. Xie and Y. Guo, "Optimisation of machining parameters in multi-pass turnings using ant colony optimisations," *International Journal of Machining and Machinability of Materials*, vol. 11, no. 2, pp. 204–220, 2012.

[17] W. H. Bangyal, "Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems," *Applied Sciences*, vol. 11, no. 16, p. 7591, 2021.

[18] S. Pervaiz, "A systematic literature review on particle swarm optimization techniques for medical diseases detection," *Computational and Mathematical Methods in Medicine*, 2021.
[19] M. C. Chen, "Optimizing machining economics models of turning operations using the scatter search approach," *International Journal of Production Research*, vol. 42, no. 13, pp. 2611–2625, 2004.
[20] S. Xu, Y. Wang, and F. Huang, "Optimization of multi-pass turning parameters through an improved flower pollination algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 89, no. 1-4, pp. 503–514, 2017.
[21] M. A. Mellal and E. J. Williams, "Cuckoo optimization algorithm for unit production cost in multi-pass turning operations," *International Journal of Advanced Manufacturing Technology*, vol. 76, no. 1, pp. 647–656, 2015.
[22] M. A. Sofuoğlu, F. H. Çakır, and S. Gürgen, "An efficient approach by adjusting bounds for heuristic optimization algorithms," *Soft Computing*, vol. 23, no. 13, pp. 5199–5212, 2019.
[23] A. R. Yildiz, "Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations," *Applied Soft Computing*, vol. 13, no. 3, pp. 1433–1439, 2013.
[24] A. R. Yildiz, "A comparative study of population-based optimization algorithms for turning operations," *Information Sciences*, vol. 210, pp. 81–88, 2012.
[25] A. R. Yildiz, "Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach," *Information Sciences*, vol. 220, pp. 399–407, 2013.
[26] A. R. Yildiz, "Optimization of multi-pass turning operations using hybrid teaching learning-based approach," *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9-12, pp. 1319–1326, 2013.
[27] A. Belloufi, M. Assas, and I. Rezgui, "Intelligent Selection Of Machining Parameters In Multipass Turnings Using Firefly Algorithm," *Modelling and Simulation in Engineering*, vol. 2014, Article ID 592627, , 2014.
[28] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. González, Ed., Springer, Berlin, Heidelberg, 2010.
[29] B. Zhu, "A Novel Quantum-Behaved Bat Algorithm with Mean Best Position Directed for Numerical Optimization," *Computational Intelligence and Neuroscience*, Article ID 6097484, 2016.
[30] X. Cai, "Bat algorithm with Gaussian walk," *International Journal of Bio-Inspired Computation*, vol. 6, no. 3, pp. 166–174, 2014.
[31] L. dos Santos Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1676–1683, 2010.
[32] X. Huang, "Gaussian Quantum Bat Algorithm with Direction of Mean Best Position for Numerical Function Optimization," *Computational intelligence and neuroscience*, vol. 2019, 2019.
[33] Y. C. Shin and Y. S. Joo, "Optimization of machining conditions with practical constraints," *International Journal of Production Research*, vol. 30, no. 12, pp. 2907–2919, 1992.
[34] W. H. Bangyal, "Detection of fake news text classification on COVID-19 using deep learning approaches," *Computational and mathematical methods in medicine*, vol. 2021, Article ID 5514220, , 2021.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 2930920, 12 pages
https://doi.org/10.1155/2022/2930920

Research Article

Investigation of Three-Dimensional Condensation Film Problem over an Inclined Rotating Disk Using a Nonlinear Autoregressive Exogenous Model

Naveed Ahmad Khan,[1] **Muhammad Sulaiman**,[1] **Ebenezer Bonyah**,[2] **Jamel Seidu**,[3] **and Fahad Sameer Alshammari**[4]

[1]*Department of Mathematics, Abdul Wali Khan University Mardan, Mardan, Khyber Pakhtunkhwa, Pakistan*
[2]*Department of Mathematics Education, Akenten Appiah-Menka University of Skills Training and Entrepreneurial Development, Kumasi, Ghana*
[3]*School of Railways and Infrastructure Development, University of Mines and Technology (UMaT) Essikado, Sekondi-Takoradi, Ghana*
[4]*Department of Mathematics, College of Science and Humanities in Alkharj, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia*

Correspondence should be addressed to Jamel Seidu; jseidu@umat.edu.gh

Received 25 December 2021; Revised 15 January 2022; Accepted 18 January 2022; Published 9 February 2022

Academic Editor: Diego Oliva

This paper analyzed the three-dimensional (3D) condensation film problem over an inclined rotating disk. The mathematical model of the problem is governed by nonlinear partial differential equations (NPDE's), which are reduced to the system of nonlinear ordinary differential equations (NODE's) using a similarity transformation. Furthermore, the system of NODEs is solved by the supervised machine learning strategy of the nonlinear autoregressive exogenous (NARX) neural network model with the Levenberg–Marquardt algorithm. The dimensionless profiles of velocity, acceleration, and temperature are investigated under the effect of variations in the Prandtl number and normalized thickness of the film. The results demonstrate that increasing the Prandtl number causes an increase in the fluid's temperature profile. The solutions obtained by the proposed algorithm are compared with the state-of-the-art techniques that show the accuracy of the approximate solutions by NARX-BLM. The mean percentage errors in the results by the proposed algorithm for $\Theta(\eta)$, $\Psi(\eta)$, $k(\eta)$, $-s(\eta)$, and $(\theta(\eta))$ are 0.0000180%, 0.000084%, 0.0000135%, 0.000075%, and 0.00026%, respectively. The values of performance indicators, such as mean square error and absolute errors, are approaching zero. Thus, it validates the worth and efficiency of the design scheme.

1. Introduction

The liquid condensate removal from cooled, saturated vapors is of immense significance in various domains of engineering, such as coating and cooling with spray, and the mechanisms of chemical vapor accumulation are widely used in the production of thin film in semiconductor industries. Many researchers have conducted a well-known study to investigate the physical model and heat transfer of the fluid with different conditions. Nusselt [1], in 1916, studied the condensation over a vertical plate that formed the basis for many researchers to study the condensation of different fluid problems. Nusselt's solution was developed by Koh et al. [2] under the consideration of convective terms, inertia, and vapor resistance in the condensation of fluid flow. The condensation of the rotating disk in steady vapor with a large volume is studied by Sparrow et al. [3]. They extended the idea of Karman V. [4] on the rotating disk, in which the Navier–Stokes equations are transformed into the set of nonlinear ordinary differential equations (NODE's) and solved numerically for the solutions corresponding to different values of finite film thickness. Becket et al. [5] and

Chary and Sarma [6] further broadened the work of Nusselt by adding vapor drag and suction on the plate.

The flow of a liquid film is made of condensing liquid on a disc and is nonlinear in nature. Generally, finding the exact and analytical solutions to such a problem is a difficult task. Different researchers have adopted various methods to find a solution for the three-dimensional condensation film problem. The governing model of the 3D flow of fluid is transformed into a set of nonlinear differential equations by Wang C. Y. [7] using the similarity transformation and solving the problem using the perturbation method. Several other techniques are used to solve the condensation film problem, such as homotopy analysis method (HAM) [8], homotopy perturbation method (HAM) [9, 10], classical Runge–Kutta and shooting method [11], extended optimal homotopy asymptotic method (EOHAM) [12], variational iteration method (VIM) [13], control volume finite element method (CVFEM) [14], differential transformation method (DTM) [15], spectral quasi linearization method (SQLM) [16], optimal homotopy analysis method (OHAM) [17], variation of parameter method (VPM) [18], and Akbari-Ganji method (AGM) [19].

In this paper, a liquid film created by the condensing fluid on a revolving disc under the centrifugal and gravitational forces is considered. A supervised learning technique is proposed to solve the system of NODE's effectively. The classical numerical approaches mostly convert the governing equations comprising partial differential equations into a discretized model that appears in the form of a set of algebraic linear or nonlinear equations. To solve a set of algebraic equations, considerable computational time and memory requirements are needed by direct solvers. In addition, such techniques are gradient-based methods with deterministic approaches. To overcome these drawbacks, artificial intelligence-based supervised learning techniques are designed that are free of gradient and only require the essential initial parameter and terminal conditions for execution. Some recent applications of the stochastic techniques include the solutions for the saturation of water and oil [20], absorption of carbon dioxide [21], the corneal model for eye surgery [22], and the temperature distribution of conductive-convective and radiative fins [23]. These facts inspire authors to explore and incorporate the intelligent strength of artificial neural networks to solve the problem formed by the condensation of 3D-fluid flow on a rotating disk. The novel contributions of the presented study are summarized as follows:

(i) The problem of a three-dimensional (3D) condensation layer over an inclined rotating disc is investigated in this article. The governing mathematical model of the problem is given by nonlinear partial differential equations (PDE's), which are transformed into the set of nonlinear ordinary differential equations (ODE's) using similarity transformations.

(ii) The dimensionless profiles of velocity, acceleration, and temperature of the problem are investigated under the effect of variations in the Prandtl number and normalized thickness by developing a supervised machine learning strategy using NARX neural networks with the backpropagated Levenberg–Marquardt algorithm.

(iii) The accuracy of the results obtained by the design algorithm is measured by comparison with the state-of-the-art techniques.

(iv) The results of mean percentage errors and performance indicators in terms of mean square error (RMSE), mean absolute deviations (MAD), absolute errors (AE), error in Nash Sutcliffe efficiency (ENSE), and Theil's inequality coefficient (TIC) are defined to validate the worth and accuracy of the design algorithm.

2. Problem Formulation

Figure 1 illustrates the rotating disk with an angular velocity Ω that is inclined at an angle β with a horizontal axis. A film of fluid with thickness t is formed by spraying on the disk with a velocity W. It is assumed that the film thickness is negligible as compared to the radius of the disk, and therefore, the end effects are ignored. $T_{\overline{w}}$ and T_0 denote the temperatures on the disk and film surface, respectively. The ambient pressure (p_0) on the film surface is assumed to be the function of z. The problem is expressed mathematically in the coordinate system (x, y, z), with the z axis being the rotation axis. The continuity, momentum, and energy equation after neglecting the viscous dissipation can be written as [7, 8] follows:

$$U_x + U_y + U_z = 0, \tag{1}$$

$$UU_x + VU_y + \overline{w}U_z - g\sin\beta = \vartheta\left(U_{xx} + U_{yy} + U_{zz}\right), \tag{2}$$

$$UV_x + VV_y + \overline{w}V_z = \vartheta\left(V_{xx} + V_{yy} + V_{zz}\right), \tag{3}$$

$$U\overline{w}_x + V\overline{w}_y + \overline{w}\overline{w}_z - \vartheta\left(\overline{w}_{xx} + \overline{w}_{yy} + \overline{w}_{zz}\right) + g\cos\beta = -\frac{P_z}{\rho}, \tag{4}$$

$$UT_x + VT_y + \overline{w}T_z = \alpha\left(T_{xx} + T_{yy} + T_{zz}\right), \tag{5}$$

U, V, and $\overline{w}$ are the components of velocity in x, y, and z directions. T is the temperature. α, ρ, and ϑ are the thermal diffusion, density, and kinematic viscosity of the fluid. For boundary conditions, zero shear stress on the surface of the film and zero slip on the disk are assumed. Thus, the boundary conditions (B.C) are given as follows:

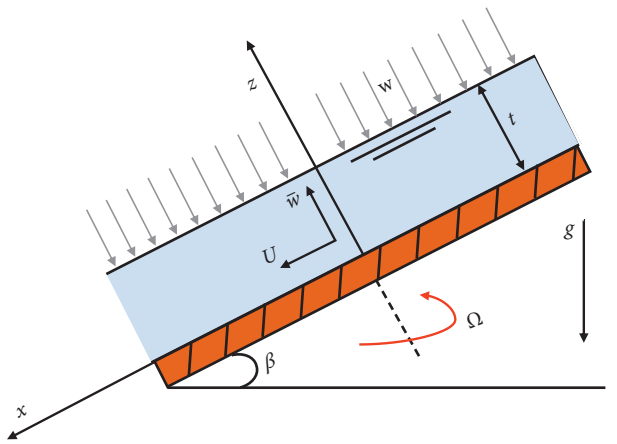


Figure 1: Geometric interpretation of condensation film problem over an inclined rotating disk.

$$\begin{aligned} T &= T_w, \\ U &= -\Omega y, \\ \overline{w} &= 0, \\ V &= \Omega x, \quad \text{at } z \\ p &= p_0, \\ U_z &= 0, \\ \overline{w} &= -W, \\ V_z &= 0, \\ T &= T_0, \quad \text{at } z \end{aligned} \tag{6}$$

In 2007, Wang C. Y. [7] introduced a transformation for the abovementioned problem, which is as follows:

$$\begin{aligned} U - \psi k(\eta)\sin\frac{\beta}{\Omega} &= -\Omega y \Psi(\eta) + \Omega x \Theta'(\eta), \\ V - gs(\eta)\sin\frac{\beta}{\Omega} &= \Omega x \Psi(\eta) + \Omega y \Theta'(\eta), \\ \overline{w} + 2\sqrt{\Omega\vartheta}\Theta(\eta) &= 0, \\ T - T_{\overline{w}} &= (T_0 - T_{\overline{w}})\theta(\eta), \end{aligned} \tag{7}$$

where η is defined as

$$\eta = z\sqrt{\frac{\Omega}{\vartheta}}, \tag{8}$$

and (1) is satisfied identically using the above transformation in (2) and (4), which can be written as

$$\Theta''' - (\Theta')^2 + \Psi^2 + 2\Theta\Theta'' = 0, \tag{9}$$

$$\psi'' - 2\Psi\Theta' + 2\Theta\Psi' = 0, \tag{10}$$

$$k'' - k\Theta' + s\Psi + 2\Theta k' + 1 = 0, \tag{11}$$

$$s'' - k\Psi - s\Theta' + 2\Theta s' = 0. \tag{12}$$

Temperature (θ) is assumed as a function of z alone, and therefore, (5) can be written as

$$\theta'' + 2\mathrm{Pr}\Theta\theta' = 0, \tag{13}$$

where $\mathrm{Pr} = \vartheta/\alpha$ is the Prandtl number. Boundary conditions for equations (9)–(13) are defined as follows:

$$\begin{aligned} \Theta(0) &= 0, \\ \Theta'(0) &= 0, \\ \Theta''(\delta) &= 0, \\ \Psi(0) &= 1, \\ \Psi'(\delta) &= 0, \\ k(0) &= 0, \\ k'(\delta) &= 0, \\ s(0) &= 0, \\ s'(\delta) &= 0, \\ \theta(0) &= 0, \\ \theta(\delta) &= 1, \end{aligned} \tag{14}$$

where $\delta = t\sqrt{\Omega/\vartheta}$ is the normalized thickness, which is also defined using the spraying velocity or condensation as,

$$\begin{aligned} \Theta(\delta) &= \frac{W}{2\sqrt{\Omega\vartheta}} \\ &= \alpha. \end{aligned} \tag{15}$$

After the flow field is found, various quantities of the fluid flow can be measured. Integrating (4) will result in the desired equation for pressure distribution of the fluid, which is given as follows:

$$\begin{aligned} p(z) = p_0 - \rho\Bigg\{ & V\left[\frac{\partial\overline{w}}{\partial z}\Big|_{z=t} - \frac{\partial\overline{w}}{\partial z}\right] \\ & + \frac{1}{2}\left[\overline{w}^2(t) - \overline{w}^2(z)\right] - g\cos\beta(z-t)\Bigg\}. \end{aligned} \tag{16}$$

If the force on the net area along x and y directions are normalized by $g\rho\sqrt{\vartheta/\Omega}\sin\beta$, then it is equal to the values of $k'(0)$ and $s'(0)$, respectively.

3. Design Methodology

3.1. Artificial Neural Networks and NARX Model. Before the 1980s, the linear parametric autoregressive (AR), the moving average (MA), and the autoregressive moving average (ARMA) were the most common approaches used by researchers to handle different types of problems [24]. These models were linear and could not be used to forecast nonlinear time-series problems. In addition, artificial neural networks (ANNs) have attracted a lot of attention because of their nonlinear and nonparametric characteristics. ANN's models are adaptive approaches based on data that can learn a system's nonlinear behavior from its historical data without having any prior knowledge of the problem. They are universal approximators for functions. Some recent applications of ANN can be found in [25]. The abovementioned articles motivate the authors to extend the idea of

ANNs to solve a nonlinear problem arising in various fields. The NARX model is a nonlinear version of the autoregressive exogenous (ARX) model that has been widely used in various applications and for modeling a variety of nonlinear dynamical systems. The NARX model is a time-series prediction model based on artificial neural networks. It learns a system's behavior more effectively than other NN's (i.e., the learning gradient method in NARX is superior). Compared to other neural networks, it converges much faster and generalizes the solutions in a much better way [26].

The multilayered perceptron (MLP) architecture underpins the NARX model [27] because of its versatility and simplicity. It is one of the most widely utilized ANN models. Input, hidden, and output layers are present in the MLP and NARX models, however, the NARX model includes the time history of the output signal as one of the inputs. The present input signal and its time history serve as the model's other inputs. The number of output neurons and variables in the problem is equal. Let $h(\cdot)$ be a nonlinear mapping function of NARX. It relates the input and output of the system by (17),

$$Y[x] = h\big(y[x-1],\ldots,y\big[x-x_y\big],u[x-1],\ldots,u\big[x-x_u\big]\big)+e[x], \tag{17}$$

$u[x]$ and $Y[x]$ represent the input and output of a system. x_u and x_y are the maximum lags for input and output, respectively. $e[x]$ is a noise or prediction error.

3.2. Learning Strategy and Performance Measures. Based on the representation of the NARX neural network model by equation (17), a data set of inputs and outputs are presented to the model during the training phase. A reference solution of 1001 points for the different cases of the 3D condensation film problem is generated using the Runge–Kutta method (RK-4) with the "NDSolve" package in mathematica. After that, an MLP is created with "nntool" in MATLAB using multiple layers of interconnected neurons with one or more hidden layers and nodes, which are connected in a feed-forward manner between the input and output layers, as shown in Figure 2. The predicted output of the multilayered perceptron is given as

$$\hat{y}_{\mathrm{MLP}} = f_2\big(W_2^T f_1(\eta)+b_2\big), \quad N = W_1^T(u)+b_1, \tag{18}$$

u is the input element of a model, b_1 and b_2 are biased terms in the hidden and output layers. W_1 and W_2 represent the synaptic weights that connect the input to the hidden and the hidden to the output layers. f_1 and f_2 represent the activation functions. In this study, the Log-Sigmoid activation function is used for neurons in the input and output layers. Once the number of weights is determined, a conventional training algorithm, such as the backpropagated Levenberg–Marquardt (BLM) algorithm, can be directly applied. To avoid the overfitting of data during the training phase, 15% of the data is reserved for cross-validation and testing. The flow chart of the problem and working strategy of the NARX-BLM algorithm is shown in Figure 3.

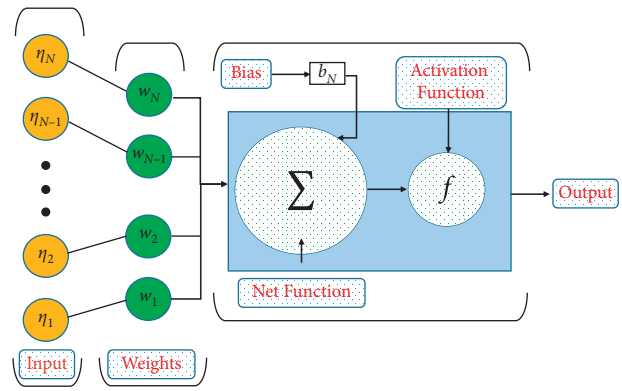


Figure 2: Details of a neuron in MLP network.

To examine the accuracy and effectiveness of the results obtained by the NARX-BLM algorithm for the 3D condensation film problem, performance indices are defined in terms of mean square error (MSE), mean absolute deviations (MAD), absolute errors (AE), root mean square error (RMSE), error in Nash Sutcliffe efficiency (ENSE), and Theil's inequality coefficient (TIC). Mathematical forms of these indices are given as follows:

$$\mathrm{MSE} = \frac{1}{k}\sum_{j=1}^{k}\left(\theta_j(t)-\widehat{\theta}_j(t)\right)^2,$$

$$\mathrm{AE} = \left|\theta_j(t)-\overline{\theta}_j(t)\right|, \qquad \left|\theta_j(t)-\overline{\theta}_j(t)\right|,$$

$$\mathrm{MAD} = \frac{1}{k}\sum_{j=1}^{k}$$

$$\mathrm{TIC} = \frac{\sqrt{(1/k)\sum_{j=1}^{k}\left(\theta_j(t)-\widehat{\theta}_j\right)^2}}{\sqrt{(1/k)\sum_{j=1}^{k}\left(\theta_j(t)\right)^2}+\sqrt{(1/k)\sum_{j=1}^{k}\left(\widehat{\theta}_j\right)^2}} \tag{19}$$

$$\mathrm{NSE} = \left\{1-\frac{\sum_{j=1}^{k}\left(\theta_j(t)-\overline{\theta}_j(t)\right)^2}{\sum_{j=1}^{k}\left(\theta_j(t)-\widehat{\theta}_j(t)\right)^2},\right.$$

$$\widehat{\theta}(x) = \frac{1}{k}\sum_{j=1}^{k}\theta_j(t), \mathrm{ENSE} = 1-\mathrm{NSE},$$

where $\overline{\theta}_j$, θ_j, and $\widehat{\theta}_j$ denote the approximate, reference, and mean solution at jth input. k denotes the number of grid points. For perfect modeling of the solutions, the desired values of AE, MAD, MSE, RMSE, and ENSE are equal to zero, while the value NSE is one.

4. Numerical Experimentation and Discussion

In this section, an artificial intelligence-based machine leaning algorithm is implemented to study the dimensionless profiles of velocity, acceleration, and temperature of the 3D condensation film problem with an inclined rotating

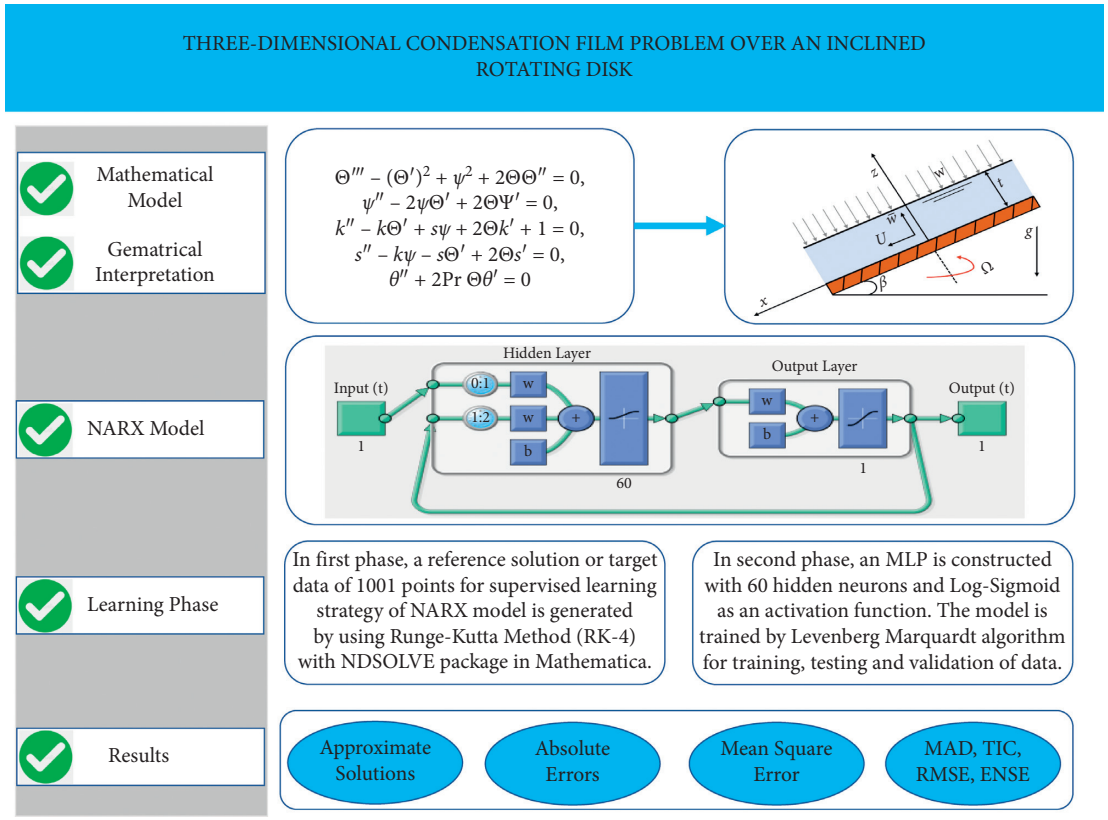


Figure 3: Governing equations of 3D condensation film problem, the NARX model, and working procedure of the proposed algorithm.

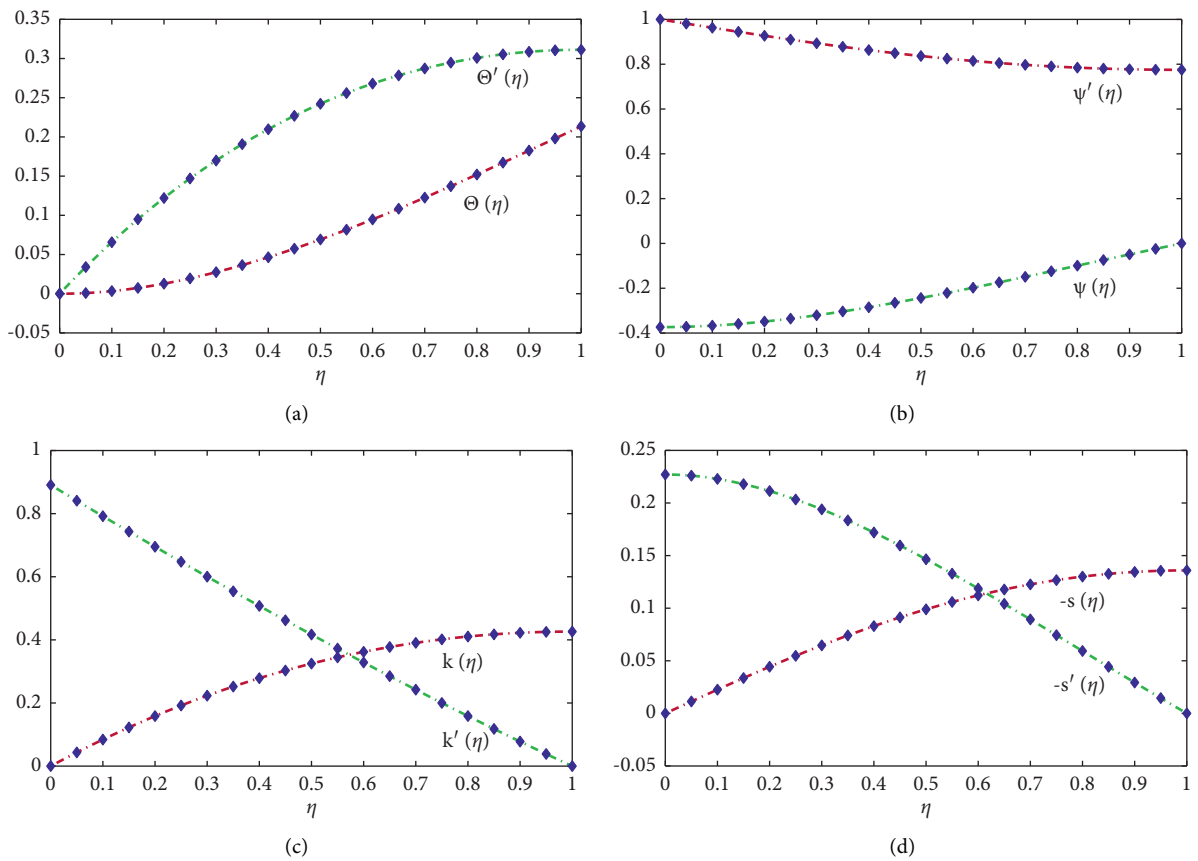


Figure 4: Continued.

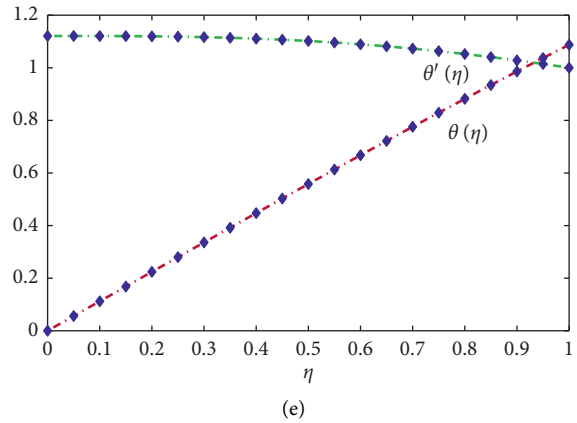


Figure 4: The change of nondimensional radial velocity and temperature profiles for condensation film problem over an inclined rotating disk with Pr = 0.7 and $\delta = 1.0$.

Table 1: Comparison of solutions obtained by the NARX-BLM algorithm with RK-4 method for Pr = 0.7 and $\delta = 0.5$.

	$\Theta(\eta)$		$\Psi(\eta)$		$k(\eta)$		$-s(\eta)$		$\theta(\eta)$	
η	Exact	NARX-BLM	Exact	NARX-BLM	Exact	NARX-BLM	Exact	NARX-BLM	Exact	NARX-BLM
0.00	0	0.000001	1	0.999993	0	0.000340	0	0.000073	0	0.000320
0.05	0.000582	0.000582	0.996086	0.996086	0.023457	0.023457	0.001995	0.001995	0.100279	0.100279
0.10	0.002247	0.002247	0.992286	0.992286	0.044418	0.044418	0.003932	0.003932	0.200554	0.200554
0.15	0.004869	0.004869	0.988700	0.988700	0.062890	0.062890	0.005760	0.005760	0.300813	0.300813
0.20	0.008328	0.008328	0.985417	0.985417	0.078877	0.078877	0.007431	0.007431	0.401036	0.401036
0.25	0.012500	0.012500	0.982514	0.982514	0.092385	0.092385	0.008910	0.008910	0.501201	0.501201
0.30	0.017266	0.017266	0.980055	0.980055	0.103421	0.103421	0.010163	0.010163	0.601278	0.601278
0.35	0.022505	0.022505	0.978090	0.978090	0.111991	0.111991	0.011163	0.011163	0.701234	0.701234
0.40	0.028099	0.028098	0.976660	0.976660	0.118102	0.118102	0.011891	0.011891	0.801032	0.801032
0.45	0.033927	0.033927	0.975792	0.975792	0.121762	0.121762	0.012333	0.012333	0.900634	0.900634
0.50	0.039873	0.039867	0.975501	0.975505	0.122980	0.122980	0.012481	0.012481	1	1

Table 2: Comparison of absolute errors in the solutions of the NARX-BLM algorithm and AG method for Pr = 0.7 and $\delta = 0.5$.

	$\Theta(\eta)$		$\Psi(\eta)$		$k(\eta)$		$-s(\eta)$		$\theta(\eta)$	
η	AGM	NARX-BLM	AGM	NARX-BLM	AGM	NARX-BLM	AGM	NARX-BLM	AGM	NARX-BLM
0.00	**0**	$1.45E-06$	0	$1.36E-05$	0	$1.86E-04$	0	$1.22E-05$	0	$6.59E-04$
0.05	$8.39E-04$	**2.57E − 07**	$1.06E-06$	**8.14E − 08**	$6.61E-06$	**2.51E − 07**	$1.99E-04$	**1.32E − 08**	$3.40E-05$	**8.63E − 07**
0.10	$1.43E-04$	**1.27E − 07**	$5.70E-07$	**4.11E − 08**	$8.13E-06$	**6.75E − 08**	$8.32E-05$	**2.52E − 08**	$5.20E-06$	**1.51E − 07**
0.15	$7.13E-05$	**6.36E − 08**	$4.44E-07$	**1.11E − 08**	$9.26E-06$	**2.70E − 08**	$6.74E-05$	**1.57E − 08**	$1.80E-07$	**8.76E − 09**
0.20	$7.77E-05$	**9.79E − 08**	$7.10E-07$	**9.70E − 09**	$1.09E-05$	**6.04E − 08**	$8.21E-05$	**9.86E − 10**	$6.47E-07$	**2.69E − 07**
0.25	$7.01E-05$	**1.59E − 08**	$7.27E-07$	**1.93E − 09**	$1.28E-05$	**8.26E − 08**	$8.61E-05$	**1.59E − 08**	$5.23E-07$	**2.10E − 07**
0.30	$6.84E-05$	**1.03E − 07**	$8.35E-07$	**2.95E − 09**	$1.48E-05$	**4.77E − 08**	$9.52E-05$	**4.60E − 09**	$6.53E-07$	**3.91E − 08**
0.35	$7.20E-05$	**1.01E − 08**	$1.00E-06$	**1.09E − 08**	$1.75E-05$	**7.38E − 08**	$1.09E-04$	**1.46E − 08**	$1.80E-07$	**2.89E − 07**
0.40	$6.32E-05$	**1.24E − 07**	$7.52E-07$	**1.56E − 08**	$1.98E-05$	**9.77E − 08**	$1.09E-04$	**2.19E − 08**	$1.61E-06$	**2.85E − 07**
0.45	$4.77E-05$	**4.97E − 07**	$1.68E-07$	**6.17E − 08**	$2.09E-05$	**2.42E − 07**	$9.93E-05$	**6.84E − 09**	$4.16E-06$	**4.96E − 08**
0.50	$6.99E-05$	**3.58E − 05**	$1.09E-06$	**4.20E − 07**	$2.57E-05$	**1.85E − 06**	$1.36E-04$	**3.65E − 07**	$2.52E-08$	$4.23E-04$

disk under the effect of variations in the Prandtl number and Normalized thickness.

Approximate solutions obtained by the proposed technique for the displacement and velocity profiles of the liquid are shown in Figure 4. Table 1 shows the comparison of the exact solutions with approximate solutions for $\Theta, \Psi, k, -s$, and θ. Table 2 shows that the results obtained by the NARX-BLM algorithm overlaps the exact and Akbari–Ganji method solutions with minimum absolute errors (AE) that lie around 3.58×10^{-05} to 1.018×10^{-08}, 1.368×10^{-05} to 1.938×10^{-09}, 1.848×10^{-04} to 2.708×10^{-08}, 1.22×10^{-05} to 9.868×10^{-10}, and 4.23×10^{-04} to 4.23×10^{-09}. The mean percentage error in the approximate solutions by the NARX-BLM algorithm are 0.0000180%, 0.000084%, 0.0000135%, 0.000075%, respectively. These facts demonstrates the accuracy of the

Table 3: Sensitivity analysis in terms of AE obtained by the proposed algorithm with different activation functions and neurons in the NARX structure for the condensation film problem with Pr = 0.7 and $\delta = 1$.

η	$\Theta(\eta)$				$\Theta'(\eta)$				$\Psi(\eta)$			
	Log-sigmoid		Tan-sigmoid		Log-sigmoid		Tan-sigmoid		Log-sigmoid		Tan-sigmoid	
	$n = 60$	$n = 30$	$n = 60$	$n = 30$	$n = 60$	$n = 30$	$n = 60$	$n = 30$	$n = 60$	$n = 30$	$n = 60$	$n = 30$
0.0	$1.17E-06$	$3.80E-07$	$6.09E-07$	$3.56E-06$	$1.24E-04$	$8.82E-05$	$1.26E-04$	$1.19E-04$	$6.83E-06$	$1.33E-05$	$1.52E-05$	$1.15E-05$
0.1	$7.51E-08$	$1.14E-08$	$7.76E-08$	$1.43E-07$	$3.11E-07$	$9.05E-07$	$1.10E-06$	$7.27E-07$	$1.62E-08$	$8.95E-08$	$5.04E-08$	$4.31E-08$
0.2	$6.19E-08$	$1.43E-08$	$4.94E-08$	$1.28E-08$	$1.49E-07$	$1.92E-07$	$4.21E-07$	$4.62E-07$	$3.41E-08$	$3.49E-08$	$6.04E-08$	$3.26E-08$
0.3	$6.34E-08$	$3.58E-09$	$8.48E-09$	$2.93E-07$	$1.02E-07$	$2.35E-07$	$2.23E-07$	$9.84E-08$	$1.23E-09$	$1.99E-08$	$3.78E-08$	$8.42E-09$
0.4	$7.02E-08$	$1.01E-08$	$1.17E-08$	$2.70E-07$	$3.75E-08$	$2.50E-07$	$1.71E-08$	$4.16E-08$	$1.38E-09$	$8.82E-09$	$7.22E-09$	$4.00E-09$
0.5	$7.57E-08$	$9.68E-09$	$2.57E-08$	$2.00E-07$	$8.89E-08$	$1.82E-07$	$2.17E-08$	$3.50E-07$	$5.61E-09$	$1.37E-08$	$6.12E-09$	$1.57E-08$
0.6	$3.20E-08$	$4.58E-09$	$1.54E-08$	$1.82E-07$	$4.79E-08$	$1.46E-07$	$1.45E-08$	$3.71E-08$	$1.61E-08$	$1.72E-08$	$7.33E-09$	$1.69E-08$
0.7	$1.42E-08$	$1.51E-08$	$3.06E-08$	$3.24E-08$	$7.27E-08$	$1.38E-07$	$1.13E-07$	$3.68E-07$	$9.91E-09$	$1.65E-08$	$2.86E-09$	$2.34E-08$
0.8	$1.97E-07$	$1.37E-08$	$7.04E-09$	$6.92E-07$	$1.53E-08$	$2.65E-08$	$1.63E-07$	$6.90E-08$	$2.78E-08$	$3.20E-08$	$2.06E-08$	$1.34E-08$
0.9	$2.91E-07$	$7.86E-09$	$1.02E-07$	$3.01E-07$	$1.15E-07$	$2.09E-07$	$3.27E-07$	$2.78E-07$	$3.95E-08$	$5.45E-08$	$6.55E-08$	$2.72E-08$
1.0	$2.87E-05$	$2.37E-05$	$4.37E-05$	$6.97E-05$	$3.68E-06$	$3.15E-06$	$4.29E-06$	$3.36E-06$	$2.76E-07$	$4.04E-07$	$5.65E-07$	$3.37E-07$

Table 4: Sensitivity analysis in terms of AE obtained by the proposed algorithm with different activation functions and neurons in the NARX structure for $k(\eta)$, $-s(\eta)$, and $\theta(\eta)$ of the condensation film problem with Pr = 0.7 and δ = 1.

η	$k(\eta)$				$-s(\eta)$				$\theta(\eta)$			
	Log-sigmoid		Tan-sigmoid		Log-sigmoid		Tan-sigmoid		Log-sigmoid		Tan-sigmoid	
	$n=60$	$n=30$	$n=60$	$n=30$	$n=60$	$n=30$	$n=60$	$n=30$	$n=60$	$n=30$	$n=60$	$n=30$
0.0	$9.79E-05$	$8.97E-05$	$2.28E-04$	$1.60E-04$	$3.71E-05$	$1.47E-04$	$1.65E-04$	$1.49E-04$	$5.08E-04$	$8.39E-04$	$9.72E-04$	$2.51E-03$
0.1	$6.46E-08$	$7.49E-07$	$3.68E-07$	$1.46E-07$	$2.16E-07$	$2.21E-07$	$1.33E-06$	$2.34E-06$	$3.29E-06$	$8.38E-06$	$7.99E-06$	$2.28E-05$
0.2	$4.94E-08$	$8.50E-08$	$5.65E-07$	$4.80E-08$	$2.25E-07$	$6.00E-07$	$6.03E-08$	$2.51E-07$	$1.34E-06$	$2.86E-06$	$2.63E-06$	$1.83E-05$
0.3	$3.00E-08$	$2.73E-07$	$1.79E-07$	$4.35E-09$	$2.23E-07$	$7.59E-08$	$2.92E-07$	$1.86E-07$	$2.05E-07$	$1.84E-08$	$4.54E-06$	$5.17E-05$
0.4	$1.70E-08$	$4.36E-08$	$8.23E-07$	$3.25E-08$	$5.91E-08$	$2.60E-07$	$1.45E-07$	$9.79E-07$	$9.90E-07$	$6.50E-07$	$2.85E-06$	$1.40E-05$
0.5	$2.05E-08$	$2.92E-08$	$2.55E-07$	$1.18E-08$	$4.18E-08$	$2.53E-07$	$2.21E-07$	$6.61E-07$	$2.39E-07$	$1.95E-07$	$3.33E-06$	$2.99E-05$
0.6	$8.47E-09$	$1.09E-07$	$2.52E-08$	$2.84E-08$	$4.07E-08$	$2.80E-07$	$3.06E-07$	$7.81E-07$	$6.40E-07$	$7.70E-08$	$2.20E-06$	$1.64E-06$
0.7	$2.48E-08$	$1.39E-07$	$6.13E-07$	$4.73E-08$	$7.41E-08$	$3.00E-07$	$3.41E-07$	$6.71E-07$	$8.51E-07$	$4.43E-07$	$1.92E-06$	$1.98E-05$
0.8	$4.69E-08$	$1.99E-07$	$4.12E-07$	$8.42E-08$	$2.53E-08$	$2.41E-07$	$3.16E-07$	$6.54E-08$	$3.21E-07$	$3.47E-08$	$4.15E-06$	$2.21E-05$
0.9	$7.91E-08$	$4.48E-07$	$1.05E-06$	$1.97E-07$	$8.91E-08$	$7.07E-07$	$7.50E-07$	$1.05E-06$	$2.81E-06$	$2.20E-07$	$1.61E-05$	$5.19E-05$
1.0	$1.04E-06$	$2.26E-06$	$5.54E-06$	$1.37E-06$	$1.07E-06$	$3.76E-06$	$3.59E-06$	$5.05E-06$	$9.58E-04$	$1.09E-03$	$1.05E-03$	$2.39E-03$

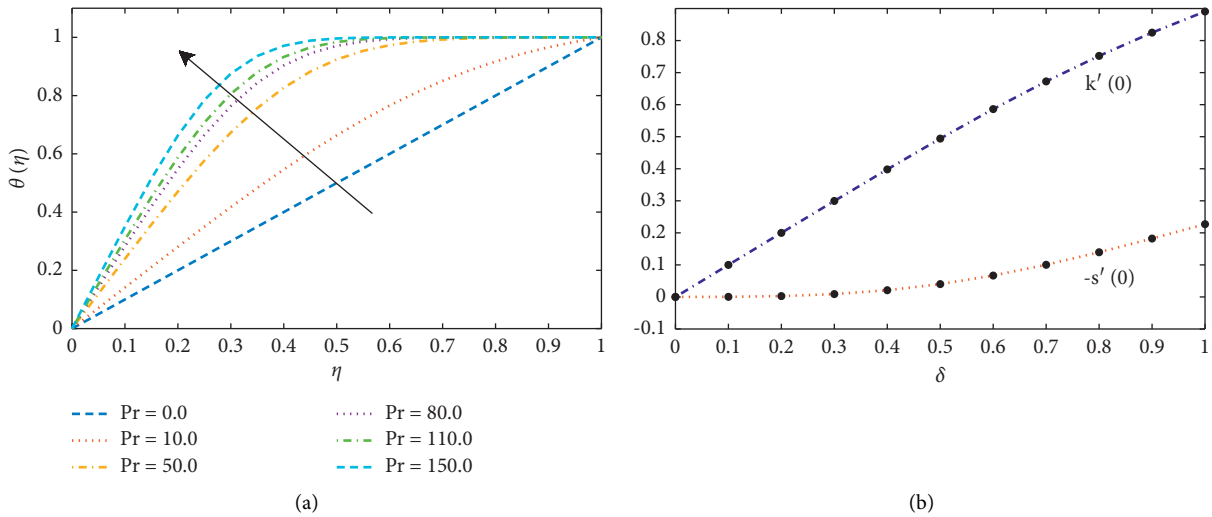


Figure 5: (a) Effect of Prandtl number on the temperature profile of the fluid. (b) Illustrates the normalized shear stress along x and y axis with different values of normalized film thickness δ and Pr = 0.7.

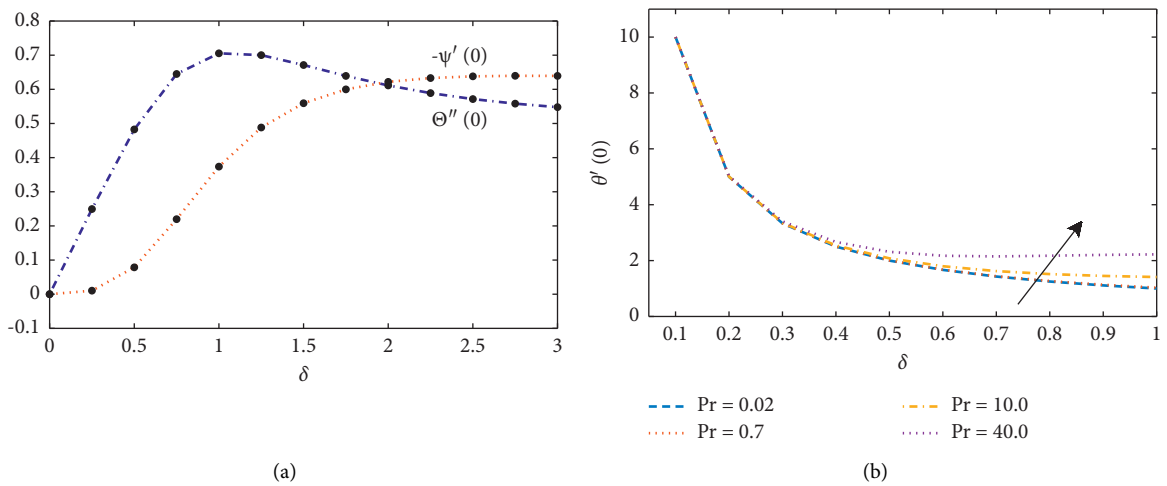


Figure 6: (a) Influence of changes in the thickness of film on $\Theta'(0)$ and $\Psi'(0)$. (b) The results of temperature gradient $\theta'(0)$ on the disk obtained by the design algorithm with Pr = 0.7 against δ.

solutions when compared with state-of-the-art techniques, such as homotopy perturbation method [9], differential transformation method (DTM) [15], and Akbari–Ganji's method [19]. The sensitivity analysis of the design algorithm in terms of different activation functions (Log-Sigmoid and Tangent Hyperbolic) and for different number of hidden neurons (n) in the NARX structure are shown in Tables 3 and 4. The results show that the convergence speed of solutions with Log-sigmoid is much higher than other activation functions.

Effect of variations in the Prandtl number on the temperature profile of the fluid are illustrated in Figure 5(a). It can be seen that the normalized temperature profile ($\theta(\eta)$) for different liquid metals starting from sodium, water, and other higher fluids increases with an increase in the Prandtl number Pr. Figure 5(b) illustrates the normalized shear stress along the x and y axis with different normalized film thickness δ. When film thickness increases, $k'(0)$ increases linearly with high intensity than $-s'(0)$. Figure 6(a) shows the results of $\Theta''(0)$ and $\Psi''(0)$ against δ. The results shows that $\Theta''(0)$ has a maximum value of 0.7085 at $\delta = 1.08$ and asymptotic to the value at 0.51023. In addition, $\Psi(0)$ possesses a minimum value at $\delta = 2.82$ and asymptotic to the value at $\delta = 6.2$.

Furthermore, to validate the efficiency, accuracy, and robustness of the proposed technique, the NARX-BLM algorithm is executed for multiple runs. The results of the mean absolute deviations (MAD), root mean square error (RMSE), Theil's inequality coefficient (TIC), and error in Nash Sutcliffe efficiency (ENSE) in terms of minimum (min), mean, and standard deviations (std.) with different activation functions and the number of neuron architecture of NARX are given in Table 5. The minimum value of MAD, ENSE, RMSE, and TIC with Log-Sigmoid function for $\Theta(\eta)$, $\Psi(\eta)$, $k(\eta)$, $-s(\eta)$, and $\theta(\eta)$ lies around 10^{-5} to 10^{-6}, 10^{-12} to 10^{-16}, 10^{-4} to 10^{-6}, 10^{-5} to 10^{-6}, and

Table 5: Statistical analysis of the results for MAD, ENSE, RMSE, and NSE obtained by multiple executions of the NARX-BLM algorithm for the condensation film problem with $\mathrm{Pr} = \delta = 1$.

	MAD			ENSE			RMSE			TIC			NSE		
	Min	Mean	Std	Min	Mean	Std	Min	Mean	Std	Min	Mean	Std	Min	Mean	Std
$\Theta(\eta)$	$5.45E-06$	$8.51E-06$	$5.30E-06$	$1.98E-12$	$6.06E-12$	$7.08E-12$	$7.26E-06$	$7.78E-06$	$9.03E-07$	$1.54E-06$	$1.65E-06$	$1.92E-07$	1	1	$7.08E-12$
$\Psi(\eta)$	$5.81E-06$	$8.65E-06$	$3.59E-06$	$5.55E-16$	$1.30E-15$	$1.01E-15$	$4.03E-06$	$5.29E-06$	$1.57E-06$	$1.32E-08$	$1.73E-08$	$5.11E-09$	1	1	$5.77E-16$
$k(\eta)$	$2.04E-05$	$3.60E-05$	$1.69E-05$	$9.19E-13$	$3.30E-12$	$2.84E-12$	$3.11E-05$	$3.86E-05$	$1.13E-05$	$1.21E-06$	$1.51E-06$	$4.43E-07$	1	1	$2.84E-12$
$-s(\eta)$	$1.47E-05$	$3.83E-05$	$2.51E-05$	$4.16E-13$	$3.61E-12$	$3.93E-12$	$2.27E-05$	$3.09E-05$	$7.48E-06$	$8.23E-07$	$1.12E-06$	$2.71E-07$	1	1	$3.93E-12$
$\theta(\eta)$	$1.00E-05$	$2.49E-04$	$2.46E-04$	$8.38E-13$	$7.23E-12$	$8.64E-12$	$2.51E-04$	$4.53E-04$	$2.58E-04$	$8.06E-07$	$1.45E-06$	$8.27E-07$	1	1	$8.64E-12$

Table 6: Results of performance function in terms of mean square error (MSE) and gradient values with different activation functions for condensation film problem with Pr = 10 and $\delta = 1.0$.

	Log-sigmoid				Tan-hyperbolic			
	Mean square error		Gradient		Mean square error		Gradient	
	$n = 60$	$n = 30$	$n = 60$	$n = 30$	$n = 60$	$n = 30$	$n = 60$	$n = 30$
Θ	$5.92E-15$	$9.32E-14$	$8.43E-11$	$2.53E-09$	$2.14E-12$	$1.50E-11$	$5.18E-10$	$5.24E-10$
Ψ	$1.35E-14$	$6.03E-14$	$5.19E-11$	$5.03E-11$	$8.44E-13$	$1.39E-14$	$1.77E-10$	$6.64E-11$
k	$6.58E-12$	$3.83E-12$	$6.40E-11$	$5.20E-08$	$3.66E-12$	$1.94E-10$	$4.12E-09$	$9.02E-10$
$-s$	$2.57E-13$	$1.48E-10$	$1.03E-10$	$3.47E-11$	$3.40E-12$	$1.85E-10$	$7.69E-10$	$6.28E-11$
θ	$1.34E-10$	$6.35E-09$	$2.46E-10$	$8.92E-09$	$1.04E-08$	$1.04E-08$	$8.93E-09$	$9.15E-09$

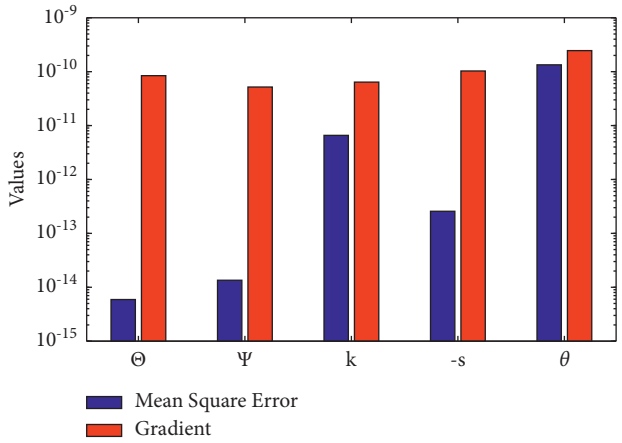


Figure 7: Mean values of performance function in term of mean square error and gradient values for condensation film problem over an inclined rotating disk with Pr = 0.7 and $\delta = 1.0$.

10^{-6} to 10^{-8}, respectively. The results of the mean square error and gradient for solutions in equations (9)–(13) are given in Table 6. It can be seen that the results obtained with log-sigmoid activation function are more accurate than the tan-hyperbolic function. The value of MSE for each function lies around 10^{-10} to 10^{-15} as shown in Figure 7.

5. Conclusion

The important outcomes of this study are summarized as follows:

(i) In this paper, we have analyzed the mathematical model of a three-dimensional (3D) condensation film problem over an inclined rotating disk by incorporating the computational strength of the supervised learning method NARX-BLM.

(ii) The designed algorithm is exploited to calculate the numerical solutions for the film problem under the influence of variations in the Prandtl number and normalized thickness.

(iii) The results demonstrate that the increase in the Prandtl number causes an increase in the temperature profile of the film. In addition, $k'(0)$ increases linearly with high intensity than $-s'(0)$ when the film thickness increases.

(iv) The results obtained by the design algorithm are compared with state-of-the-art techniques, such as the Runge–Kutta method (RK-4), homotopy perturbation method, differential transformation method (DTM), and Akbari–Ganji's method. The statistics of mean percentage error in solutions by the NARX-BLM algorithm establishes the accuracy of the design algorithm.

(v) Extensive graphical, statistical, and sensitivity analyses are conducted based on performance measures, such as MAD, ENSE, TIC, RMSE, and MSE, which show that the design algorithm is smooth, easy, and efficient for calculating the solutions to real-world problems.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. Nusselt, "Die oberflachenkondensation des wasserdamphes," *VDI-Zs*, vol. 60, p. 541, 1916.

[2] J. C. Y. Koh, E. M. Sparrow, and J. P. Hartnett, "The two phase boundary layer in laminar film condensation," *International Journal of Heat and Mass Transfer*, vol. 2, no. 1-2, pp. 69–82, 1961.

[3] E. M. Sparrow, R. Eichhorn, and J. L. Gregg, "Combined forced and free convection in a boundary layer flow," *Physics of Fluids*, vol. 2, no. 3, pp. 319–328, 1959.

[4] T. V. Kármán, "Über laminare und turbulente Reibung," *ZAMM-Journal of Applied Mathematics and Mechanics/ Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 1, no. 4, pp. 233–252, 1921.

[5] P. M. Beckett, P. C. Hudson, and G. Poots, "Laminar film condensation due to a rotating disk," *Journal of Engineering Mathematics*, vol. 7, no. 1, pp. 63–73, 1973.

[6] S. Chary and P. Sarma, "Condensation on a rotating disk with constant axial suction," *Journal. Heat Transfer*, vol. 98, no. 4, pp. 682–684, 1976.

[7] C. Y. Wang, "Condensation film on an inclined rotating disk," *Applied Mathematical Modelling*, vol. 31, no. 8, pp. 1582–1593, 2007.

[8] M. M. Rashidi and S. Dinarvand, "Purely analytic approximate solutions for steady three-dimensional problem of condensation film on inclined rotating disk by homotopy analysis method," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 4, pp. 2346–2356, 2009.

[9] M. Sheikholeslami, H. R. Ashorynejad, D. D. Ganji, and A. Yıldırım, "Homotopy perturbation method for three-dimensional problem of condensation film on inclined rotating disk," *Scientia Iranica*, vol. 19, no. 3, pp. 437–442, 2012.

[10] A. Dawar, E. Bonyah, S. Islam, A. Alshehri, and Z. Shah, "Theoretical analysis of Cu-H2O, Al2O3-H2O, and TiO2-H2O nanofluid flow past a rotating disk with velocity slip and convective conditions," *Journal of Nanomaterials*, vol. 2021, Article ID 5471813, 10 pages, 2021.

[11] N. Acharya, S. Maity, and P. K. Kundu, "Entropy generation optimization of unsteady radiative hybrid nanofluid flow over a slippery spinning disk," *Proceedings of the Institution of Mechanical Engineers-Part C: Journal of Mechanical Engineering Science*, vol. 2022, Article ID 095440622110653, 2022.

[12] H. Ullah, S. Islam, and M. Fiza, "Analytical solution for three-dimensional problem of condensation film on inclined rotating disk by extended optimal homotopy asymptotic method," *Iranian Journal of Science and Technology, Transactions of Mechanical Engineering*, vol. 40, no. 4, pp. 265–273, 2016.

[13] M. Abbasi, D. D. Ganji, I. Rahimipetroudi, and M. Khaki, "Comparative analysis of MHD boundary-layer flow of viscoelastic fluid in permeable channel with slip boundaries by using HAM, VIM, HPM," *Walailak Journal of Science and Technology*, vol. 11, pp. 551–567, 2014.

[14] M. Shutaywi and Z. Shah, "Mathematical Modeling and numerical simulation for nanofluid flow with entropy optimization," *Case Studies in Thermal Engineering*, vol. 26, Article ID 101198, 2021.

[15] M. M. Rashidi, "Analytic solution of steady three-dimensional problem of condensation film on inclined rotating disk by differential transform method," *Mathematical Problems in Engineering*, vol. 2010, Article ID 613230, 15 pages, 2010.

[16] N. Acharya, "Spectral quasi linearization simulation on the hydrothermal behavior of hybrid nanofluid spraying on an inclined spinning disk," *Partial Differential Equations in Applied Mathematics*, vol. 4, Article ID 100094, 2021.

[17] H. N. Hassan and M. M. Rashidi, "Analytical solution for three-dimensional steady flow of condensation film on inclined rotating disk by optimal homotopy analysis method," *Walailak Journal of Science and Technology*, vol. 10, pp. 479–498, 2013.

[18] O. Güngör and C. Arslantürk, "Variation of parameters method for a three-dimensional problem of condensation film on an inclined rotating disk," *Journal of Applied Mathematics and Computational Mechanics*, vol. 18, 2019.

[19] S. Berkan, S. R. Hoseini, and D. D. Ganji, "Analytical investigation of steady three-dimensional problem of condensation film on inclined rotating disk by Akbari-Ganji's methodAnalytical investigation of steady three-dimensional problem of condensation film on inclined rotating disk by Akbari-Ganji's methodretain-->," *Propulsion and Power Research*, vol. 6, no. 4, pp. 277–284, 2017.

[20] N. A. Khan, M. Sulaiman, A. J. Aljohani, P. Kumam, and H. Alrabaiah, "Analysis of multi-phase flow through porous media for imbibition phenomena by using the LeNN-WOA-NM algorithm," *IEEE Access*, vol. 8, pp. 196425–196458, 2020.

[21] N. A. Khan, M. Sulaiman, C. A. Tavera Romero, and F. K. Alarfaj, "Theoretical analysis on absorption of carbon dioxide (CO2) into solutions of phenyl glycidyl ether (PGE) using nonlinear autoregressive exogenous neural networks," *Molecules*, vol. 26, no. 19, p. 6041, 2021.

[22] W. Waseem, M. Sulaiman, A. Alhindi, and H. Alhakami, "A soft computing approach based on fractional order DPSO algorithm designed to solve the corneal model for eye surgery," *IEEE Access*, vol. 8, pp. 61576–61592, 2020.

[23] N. A. Khan, O. I. Khalaf, C. A. T. Romero, M. Sulaiman, and M. A. Bakar, "Application of euler neural networks with soft computing paradigm to solve nonlinear problems arising in heat transfer," *Entropy*, vol. 23, no. 8, p. 1053, 2021.

[24] D. S. G. Pollock, R. C. Green, and T. Nguyen, *Handbook of Time Series Analysis, Signal Processing, and Dynamics*, Elsevier, Amsterdam, Netherlands, 1999.

[25] Y. Zhang, J. Lin, Z. Hu, N. A. Khan, and M. Sulaiman, "Analysis of third-order nonlinear multi-singular emden-fowler equation by using the LeNN-WOA-NM algorithm," *IEEE Access*, vol. 9, pp. 72111–72138, 2021.

[26] T. Tsungnan Lin, B. G. Horne, P. Tino, and C. L. Giles, "Learning long-term dependencies in NARX recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, 1996.

[27] D. F. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no. 6, pp. 568–576, 1991.

Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 6390260, 10 pages
https://doi.org/10.1155/2022/6390260

Research Article

Intelligent Semantic Segmentation for Self-Driving Vehicles Using Deep Learning

Qusay Sellat,[1] SukantKishoro Bisoy,[1] Rojalina Priyadarshini,[1] Ankit Vidyarthi,[2] Sandeep Kautish,[3] and Rabindra K. Barik[4]

[1]*Department of Computer Science and Engineering, C.V. Raman Global University, Bhubaneswar, India*
[2]*Department of CSE&IT, Jaypee Institute of Information Technology, Noida, India*
[3]*LBEF Campus Kathmandu, Kathmandu, Nepal*
[4]*School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India*

Correspondence should be addressed to Sandeep Kautish; sandeep.kautish@lbef.edu.np

Received 10 November 2021; Accepted 3 January 2022; Published 17 January 2022

Academic Editor: Diego Oliva

Understanding the situation is a critical component of any self-driving system. Accurate real-time visual signal processing to create pixelwise classed pictures, also known as semantic segmentation, is critical for scenario comprehension and subsequent acceptance of this new technology. Due to the intricate interaction between pixels in each frame of the received camera data, such efficiency in terms of processing time and accuracy could not be achieved prior to recent advances in deep learning algorithms. We present an effective approach for semantic segmentation for self-driving automobiles in this study. We combine deep learning architectures like convolutional neural networks and autoencoders, as well as cutting-edge approaches like feature pyramid networks and bottleneck residual blocks, to develop our model. The CamVid dataset, which has undergone considerable data augmentation, is utilised to train and test our model. To validate the suggested model, we compare the acquired findings to various baseline models reported in the literature.

1. Introduction

Being able to move efficiently and safely in vehicles that are driverless has been a hot research topic in recent years, and many companies and research centres are trying to come up with the first completely practical driverless car model. This is a very promising field with a lot of possible benefits such as increase of safety, less costs, comfortable travel, increased mobility, and reduced environmental footprint [1]. Semantic segmentation is the process of assigning each pixel of the received image into one of the predefined classes. These classes represent the segment labels of the image, e.g., roads, cars, signs, traffic lights, or pedestrians [2]. Therefore, semantic segmentation is sometimes referred to as "pixelwise classification." The main benefit of semantic segmentation is situation understanding. It is therefore used in many fields such as autonomous driving, robotics, medical images, satellite images, precision agriculture, and facial images as a first step to achieving visual perception. Autonomous driving depends on the information received by sensors of the surrounding environment in order to form a complete picture of the driving situation. Because the visual signal is very rich in such information, doing semantic segmentation correctly is crucial for scene understanding. The more we perform semantic segmentation with a high accuracy and a short time, the more correctly the ego vehicle understands the surrounding environment and accordingly make the right decision every moment. However, semantic segmentation is challenging due to the complicated relationship between pixels in each image frame and also between successive frames. Even with the fast development of new technologies such as deep learning which have made the mission of semantic segmentation more efficient, doing accurate semantic segmentation in real

time is still a hot topic in current research as shown in detail later.

In this paper, we benefit from deep learning methods, especially convolutional neural networks (CNNs) and autoencoders (AEs) in order to design an accurate, real-time semantic segmentation model. The main contributions of this paper are as follows:

(i) Designing an accurate and real-time semantic segmentation system for self-driving cars by taking advantage of two main deep learning architectures: CNNs and AEs. A hybrid model based on the concepts of feature pyramid networks (FPNs) and bottleneck residual blocks is built to perform the semantic segmentation efficiently.

(ii) The proposed model is trained and tested using CamVid common dataset for semantic segmentation for self-driving missions. Extensive data augmentation has been done to the training set in order to overcome the problem of the small size of semantic segmentation datasets for autonomous driving.

2. Related Work

Thanks to the rapid improvements in deep learning research last decade, great results have been achieved in the field of computer vision. Developing CNNs [3] had the biggest impact on this success as tasks such as object recognition and detection have witnessed a huge jump in accuracy and speed. After the success of the early CNN models such as LeNet [4] and AlexNet [5] (AlexNet was proposed in 2012 but published in 2017), the number of proposed CNN works exploded. VGG [6], with its large number of parameters, performed well on the ImageNet dataset [7]. It increased the used number of hidden layers to 16 or 19 weight layers. At the same time, Inception [8] used the principle of network-in-network [9] to increase the depth of the CNN to 22 trainable layers. As the depth of the neural network increased, serious problems such as gradient vanishing and gradient exploding surfaced. Later proposals tried to overcome these problems by developing new techniques such as skip connections that were designed in the shape of additional connections like in ResNets [10] or the shape of concatenation connections like in DenseNets [11]. However, in addition to the gradient problems, increasing the number of layers and trainable parameters made the use of these models limited, and any idea of implementing them within constrained environments or real-time systems is impractical. Xception [12], ShuffleNet [13], MobileNetV1 [14], and MobileNetV2 provided a convenient way of designing real-time CNNs by focusing on the use of depthwise separable convolutions [15]. This allowed researchers to design mobile vision applications that are both accurate and real time. Other solutions such as EfficientNets [16] made it possible to compromise between the performance of the designed model and its complexity.

Similar to other computer vision topics, semantic segmentation research has experienced a huge improvement in the era of deep learning. In addition to CNNs, AEs were used to design semantic segmentation models that are much more efficient than old models. Recent semantic segmentation research focused on convolutional autoencoders (CAEs) which are autoencoders whose encoder and decoder parts are convolutional and deconvolutional layers, respectively. CNN models that were developed initially for object recognition and detection have been used as the backbone architectures of CAEs developed for semantic segmentation. FCN [17] used fully convolutional architecture with a large number of parameters to perform semantic segmentation. It was one of the first attempts towards getting rid of fully connected layers. SegNet [18] and SegNet-Basic [19] used VGG architecture as a backbone for the encoder and the decoder. It used the pooling indices of the encoder for the upsampling operation in the decoder. Some other architectures such as UNet [20] used some kind of skip connections between the encoder and the decoder and some other techniques such as data augmentation to increase segmentation accuracy.

Although accuracy of semantic segmentation models improved thanks to the above-mentioned models and some other architectures such as PSPNet [21], Dilated [22], and DeepLab [23], real-time semantic segmentation is still a hot research area, especially that some fields such as autonomous driving and robotics require very accurate semantic segmentation with a minimum amount of processing time. Because images are rich in semantic information, a significant number of trainable parameters are required to capture the complexity of possible images and it is very hard to develop lite segmentation models without sacrificing accuracy.

Some models such as FPN were designed with a smaller number of parameters. However, despite being efficient in the semantic segmentation missions, the encoder architecture used in the original FPN model has a structure similar to ResNets which can cause problems when generalized to work in real-time conditions. Super-lighter models such as ApesNet [24], Enet [25], ESPNet [26], and ESCNet [27] tried to minimize the number of parameters so that the semantic segmentation can be done in real time or embedded systems. Despite the fact that these models provided practical solutions to satisfy the real-time condition, crucial applications such as road scene understanding in autonomous vehicles need much more segmentation accuracy.

3. Preliminaries and Discussion

3.1. Feature Pyramid Network. The term "feature pyramid" is used in computer vision tasks to describe the process of extracting features from images in a hierarchical manner, i.e., high-level features are extracted along with low-level ones. Using this approach, a model is able to recognize objects correctly even if these objects appear in various scales. The main idea of the feature pyramid can be used to develop an end-to-end approach for semantic segmentation so that the designed algorithm produces semantically strong levels for each image. For this reason, in an end-to-end model such as ConvNets, a feature hierarchy is built layer by layer. Multiscale spatial resolutions can be built this way [28].

A feature pyramid network is a CAE that consists of two parts: bottom-up module (encoder) and top-down module (decoder). In the bottom-up module, the forward propagation of the used backbone architecture is performed. Using this backbone, a feature hierarchy is obtained. This hierarchy comprises the feature maps of several scales. A group of layers that produces output maps of the same size is said to form a "stage" so that each stage represents one pyramid level. Feature hierarchy ideas can be invested in the field of semantic segmentation by combining the low-level features with high-level features in a top-down module and skip connections. This module upsamples spatially coarser feature maps from higher pyramid levels and then tunes them with features from the bottom-up module [29, 30]. Each skip connection merges between two feature maps of the same dimensions so that one comes from the bottom-up path and the other one comes from the top-down path. Scaling ratio of 2 is used in the original FPN architecture and also in ours. So, in the bottom-up module, some stages are downsampled by a factor of 2 compared to the previous one. Also, in the top-down module, the respective stages are upsampled by a factor of 2 compared to the previous one. Figure 1 shows the architecture of feature pyramid network consisting of bottom-up and top-down modules representing feature hierarchy consisting of feature maps [31]. In addition, in some stages of the top-down path, a $(1 * 1)$ convolution is used in the skip connection to change the channel dimension of the used feature map before merging it with the coming upsampled map of the previous top-down stage in an elementwise addition manner. The channel dimension of the top-down module is referred to as a matrix of dimension d.

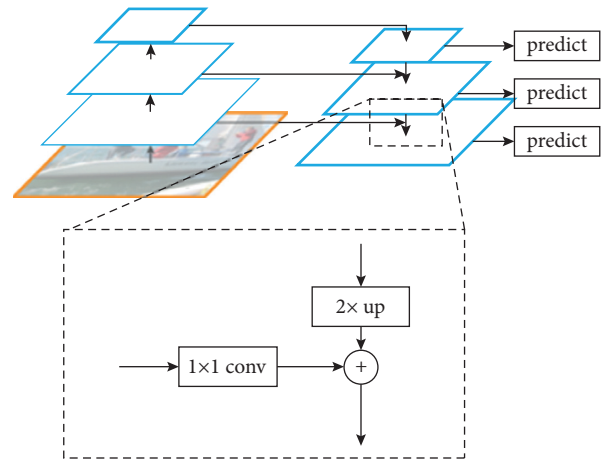


Figure 1: Architecture of feature pyramid network.

3.2. Bottleneck Residual Network. Traditional convolutional blocks have a high computational complexity that makes them hard to apply in real-time applications. One approach suggests replacing them with depthwise separable convolutions [15] that approximately have the same performance but much less complex than traditional ones. A depthwise separable convolution block is built by splitting the normal convolutional layer into two modules:

(i) A depthwise convolution layer that processes the inputs by filtering them through a $3 * 3$ convolution. This layer applies a single filter for each input channel. The re-scaling of spatial dimensions may be made in this module.

(ii) A pointwise convolution layer that combines these filtered values to create new features through a $1 * 1$ convolution. This layer combines the outputs of the depthwise convolution layer. The re-scaling of the channel dimension may be made in this module [31].

In addition, when depthwise separable convolution blocks are used to build deep neural networks, no pooling layers are used. Strides are used alternatively for downsampling tasks. Depthwise separable convolutions are used in MobileNetV1 [14]. 13 blocks of this type are used in MobileNetV1 initial configuration.

A bottleneck residual block is a slightly modified version of depthwise separable convolution blocks. A third module that is called an "expansion layer" is added. The expansion layer increases the number of channels of the input that come from the previous bottleneck block. The default expansion factor is 6. In addition, instead of a pointwise convolution layer, a bottleneck residual block has a "projection layer" that reduces the number of channels (compared to pointwise convolution layer that normally increases the number of channels). No significant change was applied to the design of the depthwise convolution layer [32–34].

In both depthwise separable convolution and bottleneck residual block, a batch normalization layer is used after each convolution process. Also, ReLU6 is used as the activation function instead of the normal ReLU. RelU6 is used as the activation function of each layer excluding the projection layer, where designers found that using a nonlinear function after this layer can make the performance worse. Additionally, in a similar manner to ResNets [10], a skip connection that links between the input of the first layer of some bottleneck residual blocks and the output of the last layer of the same block is used to overcome the problem of gradient vanishing. This connection is called a "residual connection."

Bottleneck residual blocks significantly reduce the number of computations that are needed for processing the input because of the projection layers. Because both expansion and projection layers contain learnable parameters, the useful information is transmitted to deeper layers with only little loss. Also, residual connections keep the learning process controlled and assure that useful information from the earlier layers are received by former layers. The main benefit of bottleneck residual blocks is that they reduce the amount of the data flow in the model. Because of having expansion and projection layers, the whole computation is done on an expanded (uncompressed) version of data, while the data flow between bottleneck residual blocks is minimized (a compressed version of data passes through the model).

It is worth noting that instead of using pooling layers to decrease the spatial dimensions of data as most architectures do, MobileNetV2 uses strides for spatial reduction. A stride

value $s = 2$ is used in the depthwise convolution layer when this type of reduction is needed. MobileNetV2 also does the reduction of channel dimension in the same block by making the channel dimension of the output of the projection layer less than the channel dimension of its input. In this case, no residual connection is applied to the containing bottleneck residual block as the spatial and channel dimensions of the input of the block are different from the respective dimensions of the output of the block. In case spatial and channel dimensions are preserved, a stride value $s = 1$ is used in the depthwise convolution layer and residual connection is still applied. This small difference between the two types of blocks is further explained in Figure 2. Bottleneck residual blocks are the building blocks in MobileNetV2 [31] which by default uses 17 consecutive layers of this type in its architecture (in addition to the initial convolution layer of 32 filters).

Table 1 shows more detailed specifications where h and w are the input spatial dimensions, k and k' are the numbers of input and output channels, respectively, s is the stride value, and t is the expansion factor.

4. Proposed Model Architecture

To come up with a practical semantic segmentation model for self-driving cars, it is crucial to consider two things. First, it has to be able to work in real-time conditions. Second, it has to be accurate enough so that the driverless car can depend on its results to understand the surrounding environment.

We design a model that uses the basic concepts of FPN and bottleneck residual network. The general architecture of the model is similar to FPN where we have a CAE that consists of a bottom-up path (encoder) and a top-down path (decoder) in line with the modern designs that comprise two parts: an encoder and a decoder.

However, despite being efficient in the semantic segmentation missions, the encoder architecture used in the original FPN model has a structure similar to ResNets [32] which can cause problems when generalized to work in real-time conditions.

To design an efficient model having similar working like traditional FPN model that can also perform well in real-time situations, we build the encoder part of the model so that it is similar to MobileNetV2 that is specifically designed for efficient real-time processing. As in MobileNetV2, the encoder part includes many bottleneck residual blocks (Figure 2) in a row. Here the residual connections resided in between the bottleneck layer. The input and output also consist of thin bottleneck layer. It makes use of light weight depthwise convolutional filters which help to reduce the computation time.

A detailed illustration of the architecture of our hybrid model is shown in Figure 3, where the bottleneck residual blocks are present in the left. The number of dimensions decided as presented is inferred by convolutional operators. We notice the two main parts of the model. The first one is the bottom-up part. It comprises many stages. Each stage is formed of one or more bottleneck residual blocks (this is the main difference from the original FPN model where ResNet structures instead of MobileNetV2 are used inside these stages). Similarly, the top-down path comprises many stages. The number of channel dimensions in each stage in the top-down path is $d = 256$ before halving it to $d = 128$. All the updations to the number of dimensions are performed by using the trainable parameters using convolutional operations in deep model. Skip connections link some stages in the bottom-up path with the corresponding stage in the top-down path. Finally, the outputs of all top-down stages are concatenated before the prediction process which depends on a softmax layer to predict the class label of each pixel of the input image.

4.1. Dataset. For training and testing purposes, we use Cambridge-driving Labeled Video Database (CamVid) [20, 35] which is a very common dataset for research about vision in self-driving cars. CamVid was captured from the perspective of a driving automobile. It contains video sequences of various road scenes. For semantic segmentation research purposes, a subset of these scenes was labeled at the pixel level so that each pixel was assigned to some predefined class label.

The version of the CamVid dataset we use in this research is identical to that used in modern semantic segmentation research [2, 18]. It was extracted from 5 video sequences taken at 30 Hz, so that ground truth was provided at 1 Hz (i.e., they labeled 1 frame out of 30 frames in each second). The total number of images in this version is 701: 367 images in the training set, 101 images in the validation set, and 233 images in the testing set. Each image frame has a dimension of 360480. The group of semantic classes contains 11 different classes (in addition to the void class). These classes represent the main labels of semantic segments that every road scene can normally be divided into (sky, pole, building, road, tree, sidewalk, sign symbol, car, fence, bicyclist, and pedestrian). Some samples of the CamVid dataset are shown in Figure 4. For presentation purposes, different colors were assigned to the different class labels so that each semantic segment can be visually distinguished from other segments.

Because of the nature of the captured video sequences representing realistic road scenes, there is a significant amount of imbalance in class frequencies which makes training of deep neural architectures on CamVid road scenes a big challenge for researchers. For example, classes such as sky, road, and building are 40–50 times more frequent than other classes such as bicycles and sign symbols. However, using realistic datasets such as CamVid to train and test proposed models is inevitable if we want to make our models able to be integrated into real-world self-driving systems. Because of that, we choose CamVid in our research and try to overcome the problem of class imbalance by using a common practice in the training phase of deep neural networks which is class weighting. The details of the dataset are illustrated in Table 2.

4.2. Modeling. The proposed model is implemented using Python frameworks that were designed for machine learning and computer vision tasks. The main frameworks we use are

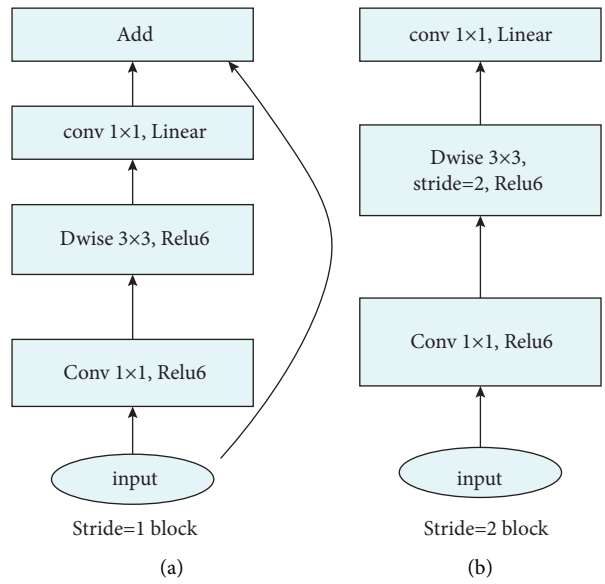


Figure 2: Types of bottleneck residual blocks (BRBs). (a) BRB does not perform spatially nor channel dimension reduction. (b) BRB performs both, and thus no residual connection is allowed.

Table 1: Details of bottleneck residual block.

Input	Operator	Output
$h \times w \times k$	1×1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3×3 dwise, $s = s$, ReLU6	$(h/s) \times (w/s) \times (tk)$
$(h/s) \times (w/s) \times tk$	linear1×1 conv2d	$(h/s) \times (w/s) \times k'$

TensorFlow, Keras, and Albumentations. The code was run on NVIDIA Tesla P100-PCIE-16 GB.

The bottom-up path of the model is similar to MobileNetV2. We use the technique of transfer learning to initialize the parameters of the bottom-up path of our model with the weight values of the original MobileNetV2 model that was trained on the ImageNet dataset [7]. The used version of the dataset is that used in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) which contains 1000 classes of different objects. By the use of transfer learning, the model is faster to learn with a high learning quality, especially when we have a relatively small dataset.

The parameters of the top-down path, on the other hand, are initialized using the normal initialization [36]. To design the deep model, the choice and selection of the optimization and hyperparameters are given as follows: (i) training by using 150 epochs with a batch size of 10, (b) learning rate to be used is 5e-4, (c) loss function is a weighted combination between Dice loss and Focal loss, and (d) optimizer to be used is RMSProp.

The first row represents the street images. The second row contains the corresponding ground truth labels of the images in the first row. The ground truth labels are nothing but the predefined class labels of the captured image as per their pixel values. The classes represent the segment labels of the image.

4.3. Training and Testing Methods. In order to get the maximum performance of our model, we have to carefully choose training and testing conditions so that the parameters of the model are trained perfectly. The model is trained on random crops from a synthetic CamVid training dataset. Each image crop has dimensions of 320×320. After that, the model is tested on image frames of full size from the CamVid testing dataset. Three training issues we focus on in our research are as follows: data augmentation, parameter initialization methods, and choosing the optimization methods and training hyperparameters.

4.3.1. Data Augmentation. Data augmentation is used to artificially expand the size of the available dataset by creating modified versions of the dataset items. If these items are image frames, like in our case, data augmentation techniques include many computer vision practices such as cropping, flipping, and so on. The main benefit of data augmentation is that it increases the diversity of the used dataset when obtaining new data is expensive in some way. Using data augmentation, the mathematical model can capture the data invariance during the training phase, and thus the resulting model has a higher ability to generalize so that it can correctly process new data. If data augmentation is used correctly, serious problems related to model training such as underfitting and overfitting can be eliminated.

As we mentioned earlier, CamVid dataset is better to be augmented so that our model can be trained better and can correctly do the semantic segmentation process on new images. To obtain high road scene diversity, the approach of data augmentation we use performs more excessive data

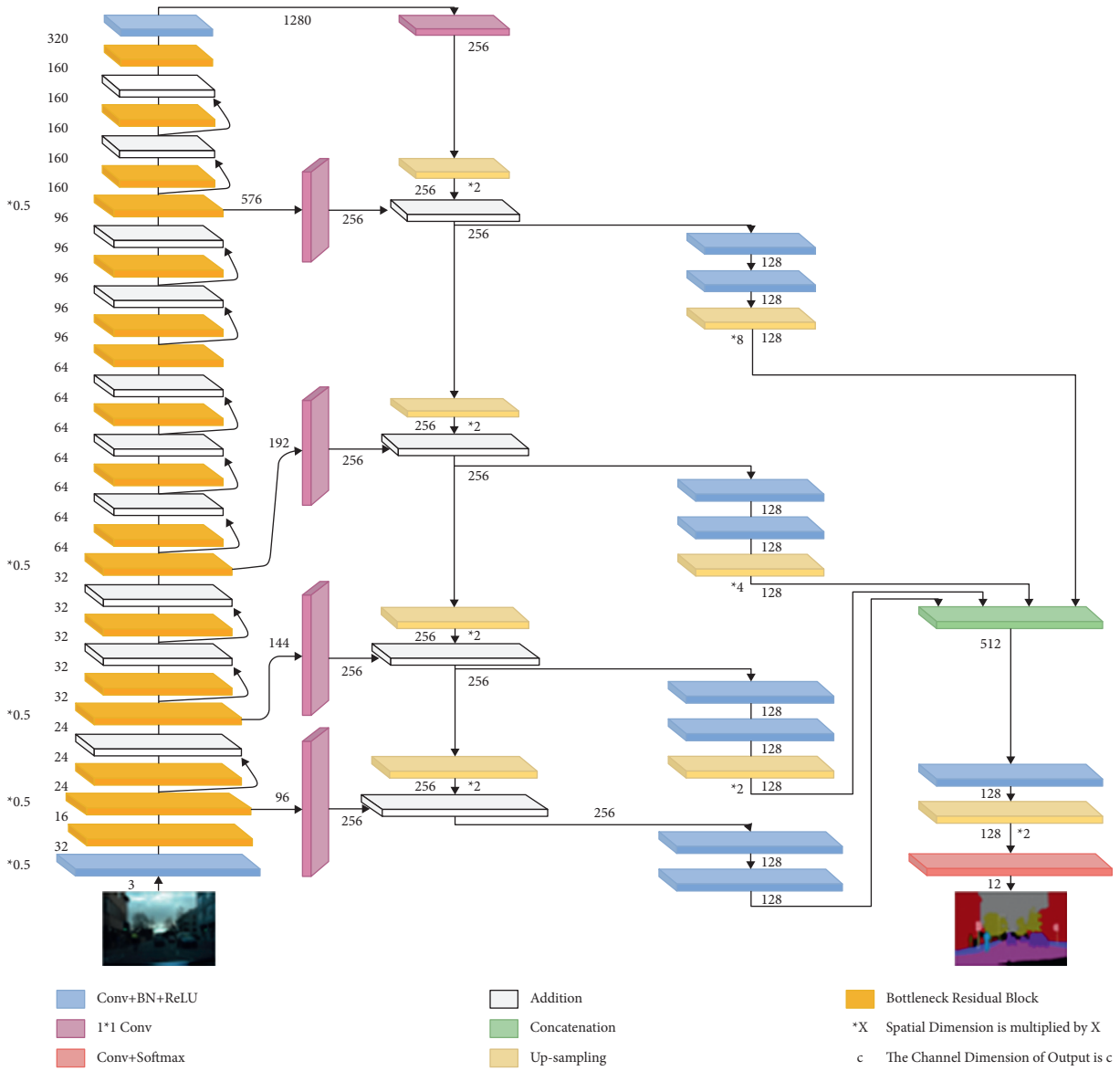


Figure 3: Proposed architecture of the bottleneck residual block.

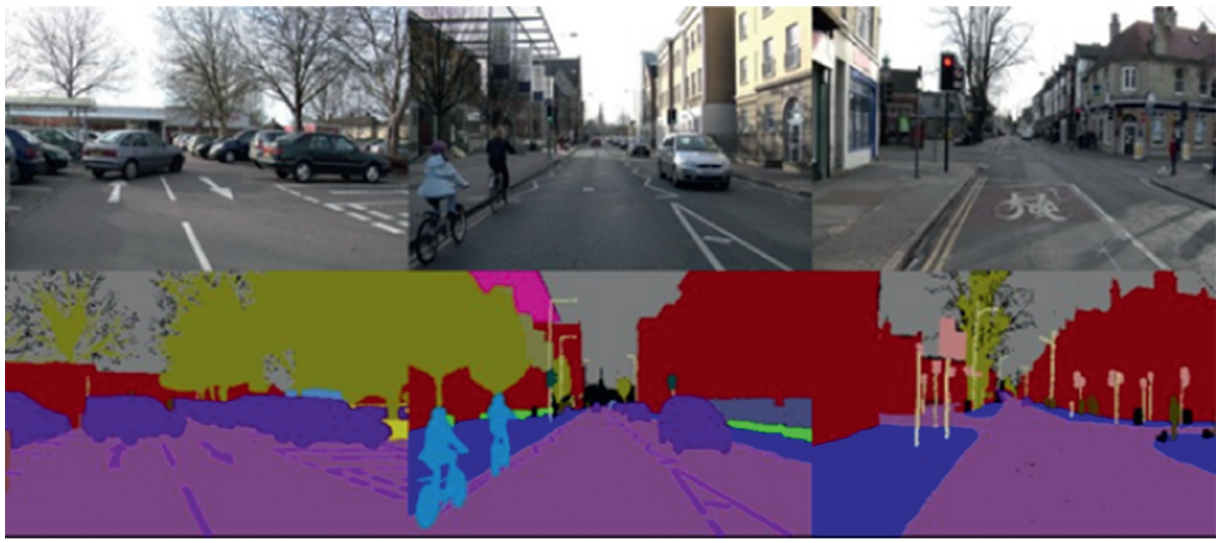


Figure 4: Some samples of the CamVid dataset [20].

Table 2: Details of the dataset utilised for validating the proposed architecture.

Dataset	Videos with object class semantic labels (it presents ground truth labels of 32 semantic classes like building, tree, sky, side walk, column-pole, fence, pedestrian, and so on)
Name	CamVid
Size	604 MB

augmentation than that used in the training process of the original UNet. The main data augmentation practices we use during the training phase include

(i) Horizontal flipping.

(ii) Shift, scale, and rotate transformations.

(iii) Random grid and optical distortions, random Gaussian noise, and random four-point perspective transform.

(iv) Histogram equalization.

(v) Random brightness and contrast, random gamma transformation, random image sharpening, and random blur (normal, motion, median, and Gaussian blur). Hue saturation need to be deleted. Random image crop with dimension 320 x 320 is used for model training.

4.3.2. Initialization. The bottom-up path of the model is similar to MobileNetV2. We use the technique of *transfer learning* to initialize the parameters of the bottom-up path of our model with the weight values of the original MobileNetV2 model that was trained on the ImageNet dataset [26]. The used version of the dataset is that used in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) which contains 1000 classes of different objects. By the use of transfer learning, the model is faster to learn with a high learning quality, especially when we have a relatively small dataset.

The parameters of the top-down path, on the other hand, are initialized using He Normal initialization (https://keras.rstudio.com/reference/initializer_he_normal.html). In this method, the weights are initialized keeping in mind the size of the previous layer which helps in attaining a global minimum of the cost function faster and more efficiently. The algorithm of initialization using this method can be summarized as follows:

(i) First, initialize weights with values taken from a standard normal distribution.

(ii) Second, multiply each weight value by $\sqrt{2}/\sqrt{n}$ where n is the number of incoming connections to the layer in which the parameter under initialization is located.

(iii) Note that bias parameters are initialized to zero.

4.3.3. Optimization and Hyperparameters. The basic design choices regarding optimization process and hyperparameters are as follows:

(i) We designed the training process to pass 150 epochs with a batch size of 10 frames (an epoch is one pass over the training dataset).

(ii) The initial learning rate is $\alpha = 5e - 4$.

(iii) The used loss function is a weighted combination between 2 loss functions: Dice loss and focal loss.

The used optimizer is RMSProp optimizer.

5. Experimental Details

The proposed model is implemented using Python frameworks that were designed for machine learning and computer vision tasks. The main libraries used are TensorFlow and Keras deep learning libraries including this Albumentationsis also being used in the work (https://albumentations.ai/). This is an efficient and user-friendly image augmentation Python library aimed at helping researchers to create fast augmentations based on a highly-optimized OpenCV library. The code was run on *NVIDIA Tesla P100-PCIE-16 GB*. The hardware specifications are presented in Figure 5. The training was done using CamVid dataset, where random image crops of 320×320 are used.

The model has been trained using the CamVid dataset followed by data augmentation. Random image crops of 320×320 were used to train the model. The model is trained for 150 epochs with an initial learning rate of $\alpha = 5e - 4$ and a batch size of 10, and the RMSProp optimizer was used. The described model is tested on the CamVid test dataset and evaluated depending on a number of evaluation metrics. When choosing metrics for evaluation, we have to take into consideration the issue of class imbalance. In other words, although using the technique of class weighting during the training process can reduce the impact of class imbalance, the resulting model usually tends to perform better on classes of higher frequency. Therefore, we choose metrics so that they do some kind of averaging between results coming from the evaluation of each class separately. In addition, another metric that is related to measuring the model complexity is used.

The following metrics are used for the present work.

(i) Mean class accuracy (mCA): in semantic segmentation, the prediction accuracy can be defined as the number of correctly classified pixels over the total number of pixels. To ensure that class imbalance does not lead to misleading accuracy values, we follow the common practice of taking the average accuracy between all calculated accuracies of the defined classes. The resulting value is called mCA.

(ii) Mean intersection over union (mIoU): this metric is common for computer vision classification tasks. In semantic segmentation tasks, intersection over union is defined as the number of pixel labels that are found in both the prediction frame and the

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.82       Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla  P100-PCIE...  off | 00000000:00:04.0 off |                    0 |
| N/A   46C   P0    34W / 250W |  12343MiB / 16280MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

Figure 5: Details of hardware specifications used in the work.

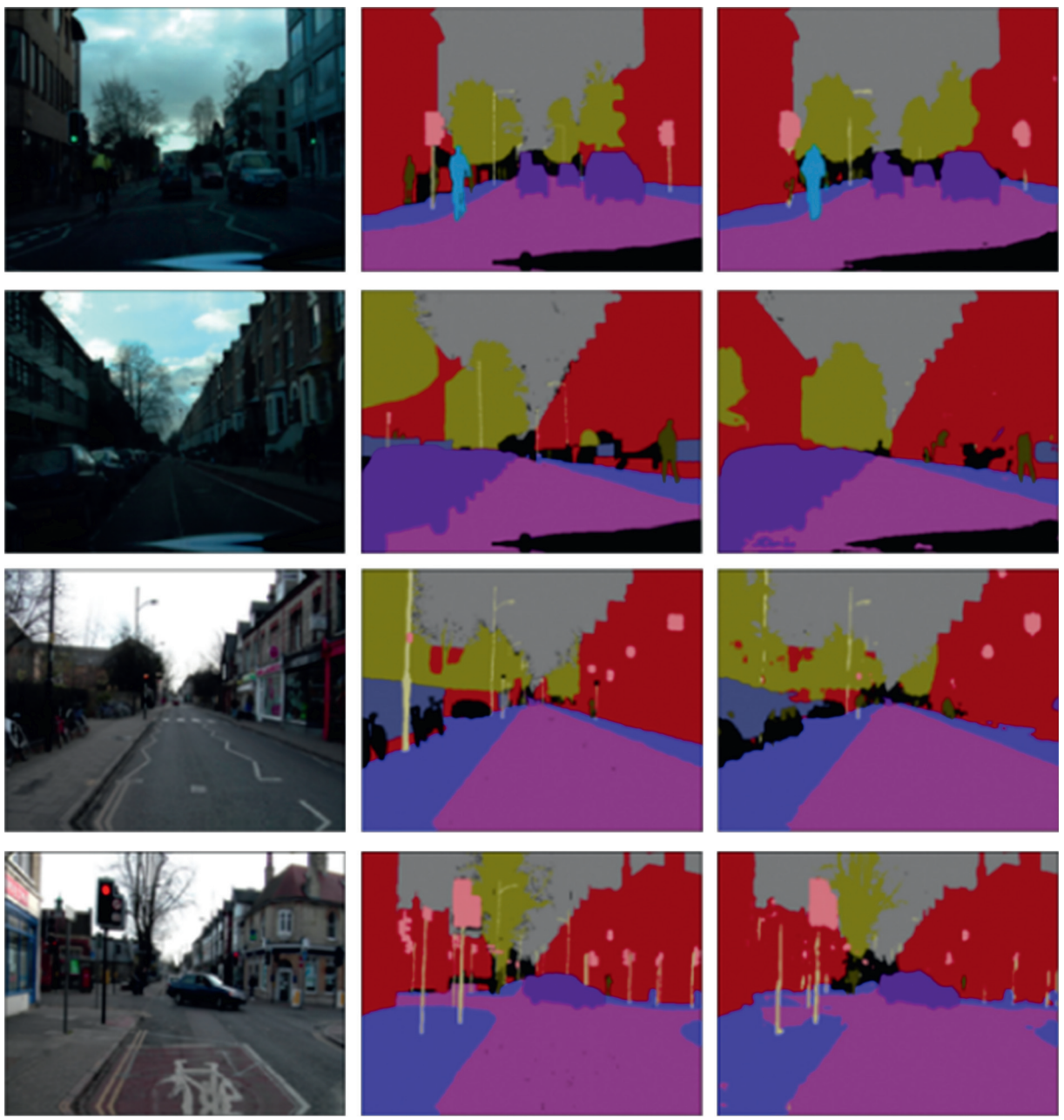


Figure 6: Samples of semantic segmentation results of the images tested on the CamVid dataset. The left column represents the original image, the column in the middle represents the ground truth labeled image, and the right column represents the predicted labels.

Table 3: Comparison between proposed model and other baseline models.

Model	mCA (%)	mIoU (%)	#Params (m)
SegNet-Basic [19]	62.9	46.2	—
SegNet [18]	65.2	55.6	29.5
FCN-8s [17]	—	57	134.5
ApesNet [24]	69.3	48	—
ENet [25]	68.3	51.3	0.36
ESPNet [26]	68.3	55.6	0.36
ESCNet [27]	70.9	56.1	0.185
DeepLab-LFOV [23]	—	61.6	37.3
Dilated-8 [22]	—	65.3	140.8
Proposed model	78.03	58.275	5.2

ground truth frame over the pixel labels that are found in either the prediction frame or the ground truth frame. IoU is calculated for each segment class separately, and then the average between all classes is taken to be the mIoU.

(iii) Number of parameters (#params): the complexity of deep learning models is very important if these models are intended to be used in real-time applications. The more complex a model is, the more the time needed for the model to calculate the output is. Therefore, the number of parameters used within a model is crucial to decide whether it is convenient to be implemented in a real-time application or not.

6. Results and Discussion

After training, the model uses the CamVid testing dataset as input and evaluates it using evaluation metrics. A mean class accuracy of 78.03% and mean intersection over union of 58.275% were obtained which makes our model a highly accurate semantic segmentation model especially with a relatively small number of parameters (5.2 M). The number of parameters used makes our model easy to be implemented as a part of a real-time autonomous system. Some samples of the semantic segmentation results of images tested on the CamVid dataset are shown in Figure 6. It represents some samples of semantic segmentation results of the images tested on the CamVid dataset used in the work. The left column represents the original image, the column in the middle represents the ground truth labeled image, and the right column represents the predicted labels.

We compare our designed model with other baseline models in terms of performance and complexity. The comparison results are shown in Table 3. As we notice, our proposed model provides a high performance with a relatively low number of parameters. The trade-off between the number of parameters and both mIoU and mCA is inevitable, and thus some models like DeepLab-LFOV and Dilated-8 would achieve a higher mIoU or mCA but are less applicable in real time due to the higher number of parameters. ENet and ESPNet have a lower number of parameters than ours but perform worse in terms of mIoU as models with a very small number of parameters may fail to capture the required complexity.

7. Conclusions

Semantic segmentation is a very important process for the perception of autonomous vehicles. It plays a major role in road scene understanding of the ego car. As deep learning methods have been improved during the last decade, more and more research is focusing on benefitting from deep learning so that better results can be obtained in all aspects of autonomous processes including perception and decision making. In this research work, we proposed a hybrid model that is built and trained upon the design principles of two deep learning models and used data augmentation techniques to increase the training quality. We aimed at making the designed architecture as accurate as real-time considerations could allow. The model was trained and tested on the CamVid dataset for which a high accuracy was obtained with a relatively small number of parameters. A comparison was done which proved the efficiency of our model compared with state-of-the-art models.

(i) In the future, more research can be conducted to increase the performance of the proposed hybrid design and overcome the current drawbacks. The main research directions that can be tested are as follows. The number of parameters of our model, despite being practical in real-time applications due to today's computing capabilities, can be made smaller by reducing the model complexity by taking into consideration the design principles of some other state-of-the-art models with a lower number of parameters.

(ii) Better performance on the mIoU metric can be a topic of future research by considering some modern design approaches of CNNs and also the design choices of models that give a high mIoU.

(iii) Other models different from MobileNetV2 can be made a backbone for the designed models. Some state-of-the-art architectures such as EfficientNet [16] represent a good candidate.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 294–305, 2019.

[2] W. Zhou, S. Lv, Q. Jiang, and L. Yu, "Deep road scene understanding," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 587–591, 2019.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, Cambridge, MA, USA, 2016.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings Of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 60, pp. 1097–1105, 2012.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[7] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[8] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, Boston, MA, USA, June 2015.

[9] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, https://arxiv.org/abs/1312.4400.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.

[12] F. Chollet, "Xception:Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258, Honolulu, HI, USA, July 2017.

[13] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, Salt Lake, UT, USA, June 2018.

[14] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[15] L. Sifre and S. Mallat, "Rigid-motion scattering for image classification," Ph. D. thesis, Ecole Polytechnique, Palaiseau, France, 2018.

[16] M. Tan and Q. V. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," 2014, https://arxiv.org/abs/1905.11946.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[19] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," 2015, https://arxiv.org/abs/1505.07293.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Munich, Germany, October 2015.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, Honolulu, HI, USA, July 2017.

[22] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, https://arxiv.org/abs/1511.07122.

[23] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFS," 2014, https://arxiv.org/abs/1412.7062.

[24] C. Wu, H. P. Cheng, S. Li, H. Li, and Y. Chen, "ApesNet: a pixel-wise efficient segmentation network for embedded devices," *IET Cyber-Physical Systems: Theory & Applications*, vol. 1, no. 1, pp. 78–85, 2016.

[25] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: a deep neural network architecture for real-time semantic segmentation," 2016, https://arxiv.org/abs/1606.02147.

[26] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 552–568, Munich, Germany, September 2018.

[27] J. Kim and Y. S. Heo, "Efficient semantic segmentation using spatio-channel dilated convolutions," *IEEE Access*, vol. 7, pp. 154239–154252, 2019.

[28] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[29] C. Deng, M. Wang, L. Liu, Y. Liu, and Y. Jiang, "Extended feature pyramid network for small object detection," *IEEE Transactions on Multimedia*, 2021.

[30] Y. Zhang, X. Sun, J. Dong, C. Chen, and Q. Lv, "GPNet: gated pyramid network for semantic segmentation," *Pattern Recognition*, vol. 115, Article ID 107940, 2021.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, San Juan, PR, USA, June 1997.

[32] D. Zhou, Q. Hou, Y. Chen, J. Feng, and S. Yan, "Rethinking bottleneck structure for efficient mobile network design," in *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference*, Springer International Publishing, Glasgow, UK, August 2020.

[33] Y. Cai, Y. Guo, S. Lang, J. Liu, and S. Hu, "Classification of hyperspectral images by spectral–spatial dense-residual network," *Journal of Applied Remote Sensing*, vol. 14, no. 3, Article ID 036513, 2020.

[34] X. Zhang, Y. Sun, Y. Wang, Z. Li, N. Li, and J. Su, "A novel effective and efficient capsule network via bottleneck residual block and automated gradual pruning," *Computers & Electrical Engineering*, vol. 80, Article ID 106481, 2019.

[35] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: a high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, Santiago, Chile, December 2015.

Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 6860503, 31 pages
https://doi.org/10.1155/2021/6860503

Research Article

Improved Sparrow Search Algorithm Based on Iterative Local Search

Shaoqiang Yan,[1] **Ping Yang**,[1] **Donglin Zhu**,[2] **Wanli Zheng**,[1] **and Fengxuan Wu**[1]

[1]*Xi'an Research Institute of High Technology, Xi'an, Shaanxi 710025, China*
[2]*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China*

Correspondence should be addressed to Shaoqiang Yan; yanshaoqiang668@163.com

Received 11 October 2021; Accepted 23 November 2021; Published 15 December 2021

Academic Editor: Ripon Chakrabortty

This paper solves the shortcomings of sparrow search algorithm in poor utilization to the current individual and lack of effective search, improves its search performance, achieves good results on 23 basic benchmark functions and CEC 2017, and effectively improves the problem that the algorithm falls into local optimal solution and has low search accuracy. This paper proposes an improved sparrow search algorithm based on iterative local search (ISSA). In the global search phase of the followers, the variable helix factor is introduced, which makes full use of the individual's opposite solution about the origin, reduces the number of individuals beyond the boundary, and ensures the algorithm has a detailed and flexible search ability. In the local search phase of the followers, an improved iterative local search strategy is adopted to increase the search accuracy and prevent the omission of the optimal solution. By adding the dimension by dimension lens learning strategy to scouters, the search range is more flexible and helps jump out of the local optimal solution by changing the focusing ability of the lens and the dynamic boundary of each dimension. Finally, the boundary control is improved to effectively utilize the individuals beyond the boundary while retaining the randomness of the individuals. The ISSA is compared with PSO, SCA, GWO, WOA, MWOA, SSA, BSSA, CSSA, and LSSA on 23 basic functions to verify the optimization performance of the algorithm. In addition, in order to further verify the optimization performance of the algorithm when the optimal solution is not 0, the above algorithms are compared in CEC 2017 test function. The simulation results show that the ISSA has good universality. Finally, this paper applies ISSA to PID parameter tuning and robot path planning, and the results show that the algorithm has good practicability and effect.

1. Introduction

With the continuous emergence of various optimization problems, various algorithms and improved algorithms are emerging [1–4]. The emergence of swarm intelligence algorithm provides new ideas for solving various optimization problems. The swarm intelligence optimization algorithm is a meta-heuristic optimization algorithm that imitates the behavior of biological populations or biological behaviors and natural phenomena in the natural world. As the optimization effect of swarm intelligence algorithms is recognized by the public, swarm intelligence algorithms develop continuously, and more and more new swarm intelligence algorithms are proposed, such as firefly algorithm (FA) [5], ant lion optimizer (ALO) [6], whale optimization algorithm (WOA) [7], sine cosine algorithm (SCA) [8], crow search algorithm (CSA) [9], Harris hawks optimization algorithm (HHO) [10], slime mould algorithm (SMA) [11], hunger games search (HGS) [12], Runge–Kutta method (RUN) [13], and colony predation algorithm (CPA) [14].

Sparrow search algorithm (SSA) is a new group intelligence optimization algorithm proposed by Xue and Shen [15] in 2020. Inspired by sparrow foraging behavior, the algorithm has obvious advantages over traditional intelligent optimization algorithms, such as grey wolf Optimizer (GWO) [16], particle swarm optimization (PSO) [17], and genetic algorithm [18], with high stability, good search accuracy, and fast convergence [19]. Despite its fast

convergence rate, the algorithm is prone to fall into local optimum and the results of optimization are random. To overcome this shortcoming, many scholars have proposed improved algorithms based on different strategies for the sparrow search algorithm and successfully solved many engineering problems. Based on the principle and model of sparrow search algorithm, Lv et al. proposed a fusion algorithm of bird swarm algorithm and sparrow search algorithm [20] and a chaotic sparrow search algorithm [21]. The former uses the search mechanism of bird swarm algorithm to add to the discoverers and followers of the sparrow search algorithm, which changes the update strategy of "full dimension reduction," effectively breaks through the local restriction of search, and strengthens the global search ability; and the latter uses tent chaotic map to initialize the population, which makes the population more uniform. After one iteration, the second iteration of chaotic disturbance and Gaussian variation is carried out according to individual fitness and average fitness, which prevents local aggregation in the optimization process, enhances its ability to jump out of local optimization, and achieves good results in the application of image segmentation. A chaotic sparrow algorithm based on cubic mapping and elite reverse learning to initialize population is presented by Tang et al. [22]. At the same time, the sinusoidal algorithm is introduced, which balances the development and exploring ability of the algorithm. At the same time, when the algorithm comes to a standstill, the Gauss Walk strategy is used to jump out of the standstill, and its optimization performance is verified in 15 benchmark functions. Finally, the UAV track planning simulation is carried out in the case of threat. Compared with other optimization algorithms, the algorithm obtains the safe and feasible track with the best cost and meets the constraints. Ouyang et al. proposed a learning sparrow algorithm [23] adding lens reverse learning during the discoverer search phase makes the search more flexible and increases the diversity of the population. A spiral guidance mechanism is introduced to make the discoverer search more precise. Then, a local search mechanism is added to prevent the omission of high-quality solutions, and compared with other swarm intelligence algorithms in 12 basic test functions and CEC 2017 test sets, this shows good optimization ability. Finally, the improved sparrow search algorithm is validated in the robot path planning, and a stable and safe optimal path is planned.

The above algorithms have made some improvements on the basis of sparrow algorithm, but there are still some shortcomings:

(1) There is still some randomness in the improved method of population initialization, which does not guarantee absolute uniformity of the population each time it is initialized.

(2) The selected improved search strategy is subject to regional limitations, is easy to exceed the boundaries, and fails to perform effective global search in the whole space, resulting in a large number of individuals exceeding the boundaries and still trapped in local optimum.

(3) By jumping directly to the discoverer, it is easy to miss the optimal solution. The improvement of local search accuracy is not significant, and there can be more improvement in search accuracy.

(4) In terms of boundary control, the strategy of updating to boundary is adopted for individuals beyond the boundary, which does not make good use of the individual location and reduces the diversity of the population.

In order to solve the above problems, the sparrow search algorithm based on iterative local search is proposed in this paper. By variable helix factor and improved iterative local search, the effective utilization and search of individuals are improved. By adding a dimension by dimension lens learning strategy and change the focusing ability of the lens, the algorithm converges faster while helping to jump out of the local optimum, and improving the boundary strategy, the population diversity is increased. To verify the optimization performance of the algorithm, ISSA PSO, SCA, GWO [16], WOA, MWOA [24], SSA, BSSA, CSSA, and LSSA are tested and analyzed on 23 basic functions. To further verify the universality of the algorithm, the above algorithms are tested and analyzed in CEC 2017 test function [25,26]. Finally, ISSA is applied to PID parameter tuning [27]. The accuracy and convergence speed of the tuned results are improved compared with the SSA, which shows that the algorithm has good practicability. The main contributions of this paper are as follows:

(1) In order to improve the effective use and search of individuals, a variable helix factor strategy is proposed and boundary control is improved

(2) An improved iterative local search strategy is presented to improve the problem of low accuracy and missing better solutions in the search process

(3) In order to improve the ability of the algorithm to jump out of local optimization, a dimension by dimension lens learning method is proposed to change the lens focusing ability

(4) The versatility and flexibility of the algorithm using benchmark functions and CEC 2017 functions are validated

(5) ISSA is used to optimize PID parameters to help quickly complete PID parameter tuning

(6) ISSA is used to optimize the robot path planning problem and help get fast and stable results

The main work arrangement of this paper is as follows: Section 2 introduces the basic sparrow algorithm. Section 3 introduces and analyses the ISSA. Section 4 compares and analyzes the algorithm on the basic test function. Section 5 compares and analyzes the algorithm on CEC 2017. Section 6 applies the algorithm to PID parameter tuning. Section 7 applies the algorithm to robot path planning. Section 8 discusses and provides future research directions.

2. Sparrow Search Algorithm

In the process of sparrow foraging, there are two behavioral strategies: discoverer and follower. The individuals with better positions in the sparrow population generally take 10%–20% of the total population as discoverers, while the remaining individuals take part in the process. At the same time, 10%–20% of individuals are randomly assigned as scouters. The discoverer is responsible for leading the population in search direction and finding food, while the follower follows the discoverer to obtain food, and the scouter is alert to environmental threats and warns the sparrow population to move closer to a safe area.

In order to describe the process of sparrow foraging through mathematical models, it is necessary to formulate rules to simplify various behaviors of sparrows. The specific rules are as follows:

(1) The individual energy of sparrow population depends on individual fitness evaluation, and the individual energy of discoverer is higher than that of discovers.

(2) Once the scouters in the sparrow population find the threat of the external environment, they begin to send out an alarm signal. When the alert value is greater than the security threshold, the discoverers direct the population to the security zone.

(3) Sparrows have flexible individual behavior strategies and can switch between discoverers and followers. As long as the individual energy reaches a certain degree, they can become discoverers, but the proportion between discoverers and followers in the population remains unchanged.

(4) Sparrows with low energy may fly to other places for feeding in order to obtain higher energy.

(5) When there is an external environmental threat, the sparrows at the edge of the population will quickly move to the safe area, and the sparrows in the middle of the population will immediately swim away to get close to other sparrows.

The discoverer is responsible for guiding the population to forage or to the location of the safe zone. The location update is described below:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\dfrac{-i}{\alpha \cdot M}\right), & \text{if } R_2 < \text{ST}, \\ X_{i,j}^{t} + Q \cdot L, & \text{if } R_2 \geq \text{ST}. \end{cases} \tag{1}$$

Among them, t represents the current number of iterations, and M is the maximum number of iterations. $X_{i,j}$ represents the current position of the i-th sparrow in the j-th dimension. $\alpha \in [0, 1]$ and is a random number. R_2 represents an early warning value, ST is the security threshold, and $R_2 \in [0, 1]$, $\text{ST} \in [0.5, 1]$. Q represents a random number that follows a normal distribution. L represents a $1 \times d$ matrix with all elements 1. When $R_2 < \text{ST}$, this indicates that the population environment is safe at this time, no predators are found around them, and the discoverers can conduct extensive searches to guide the population to higher energy levels. When $R_2 \geq \text{ST}$, this indicates that an individual within a population has discovered a predator and issued an alert, and that the discoverer quickly adjusts the search strategy to flee the current location, leading the population to a safe location.

In order to obtain high-quality food, followers follow the discoverer or forage alone, so the location of followers is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{\text{worst}}^{t} - X_{i,j}^{t}}{i^2}\right), & \text{if } i > \dfrac{n}{2}, \\ X_{P}^{t+1} + \left|X_{i,j}^{t} - X_{P}^{t+1}\right| \cdot A^{+} \cdot L, & \text{otherwise}. \end{cases} \tag{2}$$

Among them, X_p is the best position currently occupied by the discoverer, and X_{worst} represents the worst position currently. A is a $1 \times d$ matrix with only 1 or -1 elements, where $A^{+} = A^{T}(AA^{T})^{-1}$. When $i > n/2$, this indicates that the less adaptable 1st participant is not getting food, is very hungry, and needs to fly elsewhere to get more energy; when $i < n/2$, followers monitor the finder and compete for food with the finder with a higher predator, thereby increasing their energy.

When aware of the danger, the sparrow population will make antipredation behavior, and its mathematical expression is as follows

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \cdot \left|X_{i,j}^{t} - X_{\text{best}}^{t}\right|, & \text{if } f_i \neq f_g, \\ X_{i,j}^{t} + K \cdot \left(\dfrac{\left|X_{i,j}^{t} - X_{\text{worst}}^{t}\right|}{(f_i - f_w) + \varepsilon}\right), & \text{if } f_i = f_g, \end{cases} \tag{3}$$

where X_{best} represents the current global optimal position. β is the control step parameter and is a normally distributed random number with a mean value of 0 and a variance of 1. K is a random number belonging to $[-1, 1]$, which controls the direction of the sparrow's movement as well as the step. f_i represents the fitness value of the current sparrow individual. f_g and f_w are the current optimal and worst fitness values, respectively. ε is a very small real number that prevents the denominator from being zero. When $f_i \neq f_g$, this indicates that the current sparrow is at the edge of the population and that an individual is vulnerable to predators, and it is necessary to approach other individuals in the population center to reduce the risk of predation. When $f_i = f_g$, this indicates that individuals in the center of the population are aware of the danger and need to flee from their current location in order to avoid it.

3. Sparrow Search Algorithm Based on Iterative Local Search (ISSA)

3.1. Variable Helix Factor. Followers occupy the majority of individuals in the population. When $i < (N/2)$, they have a unique update mechanism that draw closer to the

discoverers quickly; it results in a fast convergence rate. When $i > (N/2)$, they have the ability to search globally. However, the global search ability is not strong, limited by the boundary of the search area, which tends to cause aggregation at the boundary in the early stage, resulting in loss of population diversity, easy to fall into the local extreme phenomenon, and poor ability to jump out of the local optimal.

The location update method of follower adopts the random coefficient that obeys the normal distribution. Without considering the boundary, the coefficient has strong global search ability. However, when considering the boundary, this mechanism is detrimental to the individuals who are at or near the boundary. Many absolute coefficients exceed 1. When the boundary is exceeded, it causes the population individuals to aggregate at the boundary, does not make full use of the current location, and results in a significant decrease in population diversity and overall algorithm performance. Based on this, a variable helix factor is proposed to reduce the number of individuals beyond the boundary, control the search step and direction, make full use of the whole population space, the space for early search is large, maintain the diversity of the population, and help jump out of the trap of local optimum. Later local search is more detailed, which greatly improves the search ability of the algorithm, as shown in Figure 1.

The formula for the variable helix factor works as follows:

$$H = a \cdot \cos(k \cdot l \cdot \pi), \tag{4}$$

$$a = \begin{cases} 1, & t < \dfrac{M}{2}, \\ e^{5 \cdot l}, & \text{otherwise}, \end{cases} \tag{5}$$

$$l = 1 - 2 \cdot \frac{t}{M}. \tag{6}$$

H is a variable helix factor, a is a parameter used to control the helix, with a value of 1 in the earlier period and a decreasing number of iterations in the later period; k is a parameter representing the helix cycle, $(M/10)$ in general; and l is a parameter that decreases linearly from 1 to −1 in terms of the number of iterations.

In this paper, we select a high-dimensional test function, step function, and a low-dimensional test function, Shekel function, to test the original model and the improved model for individuals beyond the boundary, as shown in Figure 2. Set the population individual as 30 and the number of iterations as 500, run each for 10 times, and take the average value. The test results are shown in Table 1.

According to Table 1, when the followers conduct extensive search, the number beyond the boundary accounts for the vast majority of the total number beyond the boundary of the sparrow population, resulting in the loss of a large number of individuals. Therefore, it is necessary to restrict the extensive search scope of the followers; based on the variable helix factor, the number of times the followers exceeding the boundary is greatly reduced to 0, which fully retains the favorable position information of the current individual.

Improvements to the extensive search of the followers make it possible for the followers to make full use of the whole search space, get rid of the attraction of the local optimal solution more easily, strengthen the search for the whole space, maintain the diversity of the population, enhance the ability of algorithm exploration in the early stage, and enhance the ability of algorithm development in the later stage. Based on this, the formula is updated as follows:

$$x_{i,j}^{t+1} = \begin{cases} \cos(a \cdot l \cdot \pi) \cdot \exp\left(\dfrac{x_{\text{worst}}^t - x_{i,j}^t}{i^2}\right), & i > \dfrac{N}{2} \text{ and } t < \dfrac{M}{2}, \\ e^{5 \cdot l} \cdot \cos(a \cdot l \cdot \pi) \cdot \exp\left(\dfrac{x_{\text{worst}}^t - x_{i,j}^t}{i^2}\right), & i > \dfrac{N}{2} \text{ and } t > \dfrac{M}{2}. \end{cases} \tag{7}$$

3.2. Improved Iterative Local Search. When $i < (N/2)$, the followers have a unique update mechanism that quickly closes to the discoverer's optimal solution, which results in fast convergence of the algorithm. The followers jump directly to the neighborhood of the discoverer's optimal solution. Although they have some development ability near the current optimal solution, they do not make enough use of the current solution and have poor stability. They cannot guarantee the quality and accuracy of the solution and have poor local development ability. Once trapped and unable to jump out of the local extreme state, the overall performance of the algorithm will be limited. Inspired by [28–30], this paper presents an improved iterative local search.

Local search algorithm [31] is a simple greedy search algorithm that is improved from the hill-climbing method. Local search starts from an initial solution, then searches the neighborhood of the solution, and updates the solution if there is a better solution or returns to the current solution. Iterative local search is an exploratory method that adds perturbation to the local optimal solution obtained by local search and then re-searches the local solution.

The improved iterative local search first performs a local search near the initial solution, then disturbs the initial solution by updating the location of the followers closer to the discoverer, and then searches the updated location again. It makes full use of the location information of the current individual

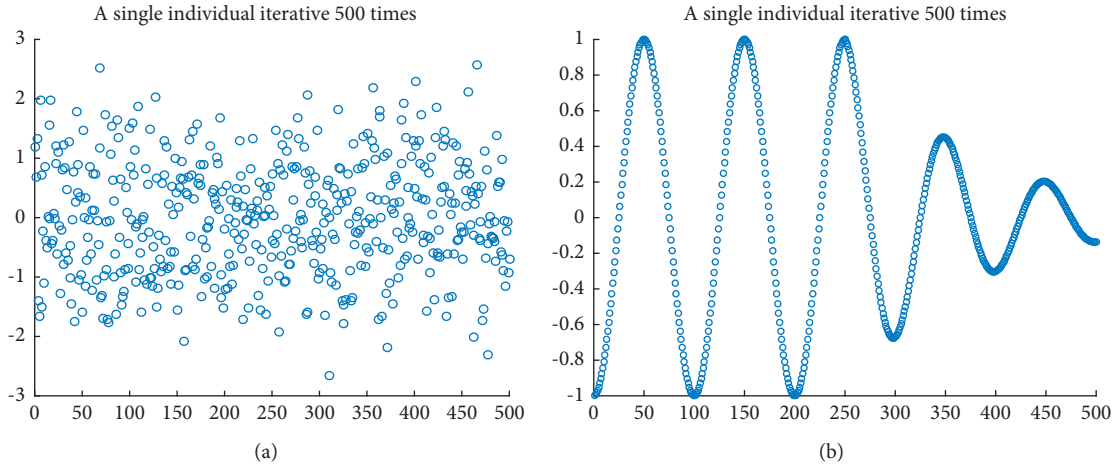


Figure 1: Coefficient model. (a) Original random coefficient. (b) Variable helix factor.

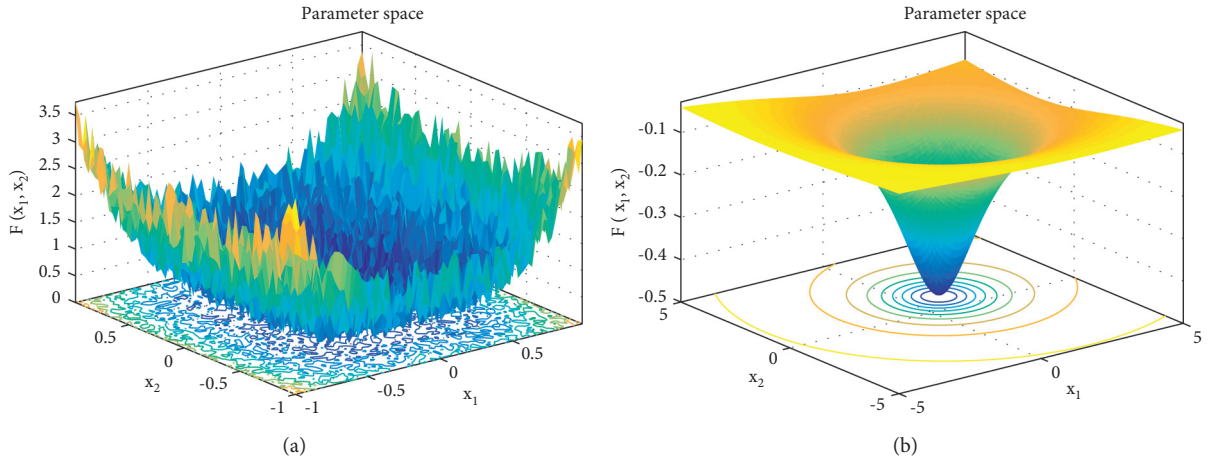


Figure 2: Parameter space. (a) Step function. (b) Shekel function.

Table 1: Statistics on the number of individuals beyond the boundary.

Function	Step	Shekel
Dimension	30 dim	4 dim
Boundary	[−1.28 1.28]	[0 10]
Number of times the original algorithm follower exceeded the boundary	1505	3739
Number of times the follower exceeded boundaries after improvement	0	0
Total number of times the original algorithm exceeded the boundary	1696	4381
Total number of times the improved algorithm exceeded the boundary	161	1095

and the location information of the current optimal solution to make the search more flexible. The first modification uses the mechanism by which the follower approaches the discoverer to disturb, so that the individual jumps near the current global optimal position to jump out of the local optimal position; the second modification uses the local search to obtain stable optimization results and improve the accuracy of the local search; and the third modification uses the local search result to obtain the optimal local search result to ensure the quality of the solution, as shown in Figure 3.

At the same time, two cases are illustrated in this paper. Figure 3(a) indicates that after individual disturbance, a better solution can be found near the current global optimal solution; and Figure 3(b) indicates that an individual can find a better solution using his own favorable position, both of which help to jump out of the local optimal solution. This paper has not been replaced by the current global optimal solution to maintain population diversity and prevent premature convergence.

3.2.1. The First Modification (SSA Method). The improved iterative local search strategy is more suitable for this algorithm than the original iterative local search. It works by first disturbing the initial solution to get an intermediate solution (when $i < n/2$, the followers of the SSA are disturbed

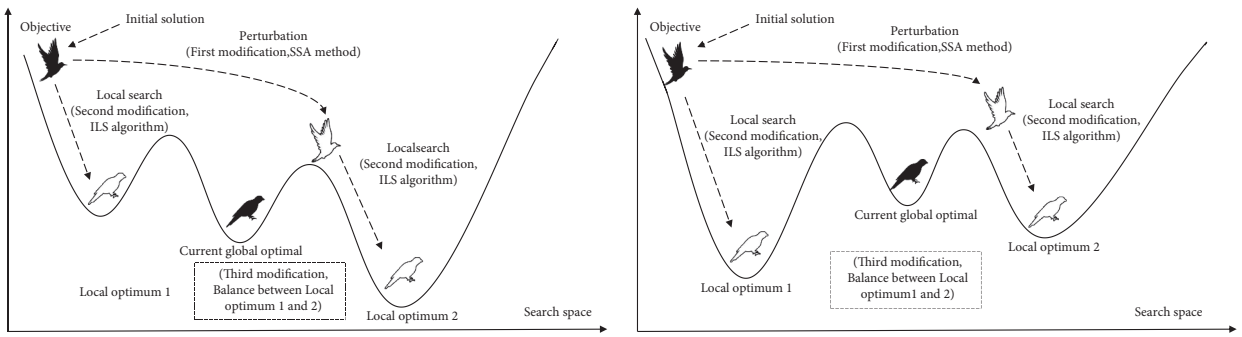


Figure 3: Main idea of improved iterative local search.

by a unique update mechanism that closes quickly to the finder's optimal solution). The initial and intermediate solutions are then searched again for a better solution. The first modification is used to solve the problem that the SSA is easy to fall into local optimal solution. The algorithm flow is as follows:

Step 1. The initial solution X^* is perturbed by the current optimal solution X_{best} to obtain the intermediate solution X^{**}. The perturbation formula is as follows:

$$X^{**} = X_{\text{best}} + \left|X^* - X_{\text{best}}\right| \cdot A^+ \cdot L. \tag{8}$$

The formula for the original followers is as follows:

$$x_{i,j}^{t+1} = x_P^{t+1} + \left|x_{i,j}^t - x_P^{t+1}\right| \cdot A^+ \cdot L, \quad i < \frac{N}{2}, \tag{9}$$

X_{best} is x_P^{t+1}, which is the best position the discoverer currently occupies; X^* is $x_{i,j}^t$, which is the current position; X^{**} is $x_{i,j}^{t+1}$, which is the updated location.

3.2.2. The Second Modification (ILS Algorithm). The second modification solves the problem of unstable and inaccurate optimization results of the SSA. The ILS algorithm here represents the local search stage in the ILS algorithm. It works by searching the initial and intermediate solutions locally to get a better solution and effectively utilizes the current position of the initial solution to prevent the individual from missing the better solution in the process of jumping directly to the current optimal solution. At the same time, local search near the current optimal solution helps to improve the accuracy of the solution and jump out of the local optimum in a small range. The algorithm flow is as follows:

Step 2. Initial solution X^* starts local search, and the formula is as follows:

$$X^1 = X^* \cdot \text{rand}(), \tag{10}$$

rand() is a random number between 0 and 1.

Step 3. Intermediate solution X^{**} starts local search, and the formula is as follows:

$$X^2 = X^{**} \cdot \text{rand}(). \tag{11}$$

3.2.3. The Third Modification. The third modification is to optimize the local search results to ensure the quality of the solution. The working principle is to use greedy strategy to compare the local search results of the initial and intermediate solutions and select the better value as the final solution X. The algorithm flow is as follows:

Step 4. Calculate fitness $f(X^1)$ of X^1.

Step 5. Calculate fitness $f(X^2)$ of X^2

Step 6. Compare $f(X^1)$ and $f(X^2)$ to select the best individuals for location updates; that is,

If $f(X^1) \le f(X^2)$
$X = X^1$
else $X = X^2$
end

Based on the SSA and the three modifications above, the formula of the follower ($i < N/2$) in the SSA is updated as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \cdot \text{rand}(), & i < \dfrac{N}{2} \text{ and } f\left(X^1\right) \le f\left(X^2\right), \\ \left(x_p^{t+1} + \left|x_{i,j}^t - x_p^{t+1}\right| \cdot A^+ \cdot L\right) \cdot \text{rand}(), & i < \dfrac{N}{2} \text{ and } f\left(X^1\right) > f\left(X^2\right). \end{cases} \tag{12}$$

3.3. Dimension by Dimension Lens Imaging Learning. Swarm intelligence algorithms have the disadvantage of easily falling into local optimum. In this regard, relevant scholars have proposed the method of adding reverse learning to swarm intelligence algorithm [32–34]. The solution after reverse learning can be closer to the optimal solution. Generally, reverse learning can only search for the optimal solution in a certain space, but it still has monotonicity and the possibility of falling into local optimum. Lens learning [35, 36] has better optimization ability than general direction learning and can continuously converge to the optimal solution in a certain space. However, once there is no optimal solution in the selected space, it will still lead to local optimization in the end. At the same time, scouters have antipredator behavior to help the population to jump out of the local optimum, but their ability to jump out of the local optimum is unstable, resulting in sometimes unable to jump out of the local optimum. In view of this phenomenon, this paper proposes a dimension by dimension lens imaging learning strategy to change the focusing ability of the lens, which is used to strengthen the scouters' ability to jump out of the local optimum, lens learning for each dimension, and reduce the mutual interference between the dimensions. In the early stage, the lens with poor focusing ability is selected for reverse learning. At this time, the imaging is divergent and far away from the lens, which can help jump out of the local optimum, as shown in Figure 4(a); in the later stage, the lens with strong focusing ability is selected for reverse learning. The imaging is concentrated and close to the lens, which can accelerate the convergence, as shown in Figure 4(b). Comparison of three kinds of reverse learning is shown in Table 2.

The principle of lens imaging is as follows:

As shown in Figure 4, taking one-dimensional space as an example, it is assumed that there is an individual with a height of h at the position. Under the action of the lens, it forms an image with a height of h^* at the position x^*. a, b is the boundary, and the lens position is the midpoint of $[a, b]$. According to the principle of lens imaging and triangle similarity principle:

$$\frac{((a+b)/2)-x}{x^*-((a+b)/2)}=\frac{h}{h^*}=k. \tag{13}$$

By transforming the above formula, we can get

$$x^*=\frac{a+b}{2}+\frac{a+b}{2k}-\frac{x}{k}, \tag{14}$$

k is used to indicate the focusing ability of the lens, that is, the imaging size. When $k=1$, it can be simplified as follows:

$$x^*=a+b-x. \tag{15}$$

This is the general reverse learning strategy. It can be seen that the general reverse learning strategy is a special case of lens imaging. The general reverse learning strategy k is fixed, and the obtained individuals are also fixed. Lens learning can change the position of individuals by adjusting k, so as to further enhance the diversity of groups. Generally, it takes a constant that is not equal to 1. This paper proposes a strategy of linearly increasing K according to the number of iterations, that is

$$k=a+\frac{t}{M}, \tag{16}$$

where a is a small constant to prevent the previous iterative imaging from being too large, which is taken as 0.1 in this paper. The early k is small, and the imaging is large; in the later stage, k is near 1, and the imaging is slightly smaller, which can help convergence.

At the same time, lens imaging is extended to each dimension, and lens imaging reverse learning is performed for each dimension. The formula is extended as follows:

$$x_j^*=\frac{a_j+b_j}{2}+\frac{a_j+b_j}{2k}-\frac{x_j}{k}, \tag{17}$$

where j is the current dimension, a_j is the lower bound of the j-th dimension, and b_j is the upper bound of the j-th dimension. At the same time, this paper adopts dynamic boundary:

$$\begin{aligned} a_j&=\min\left(x_j\right),\\ b_j&=\max\left(x_j\right), \end{aligned} \tag{18}$$

where $\min(x_j)$ is the minimum value of the j-th dimension in all individuals, and $\max(x_j)$ is the maximum value of the j-th dimension in all individuals. Because a_j and b_j do not represent the boundary of the whole search space, when the imaging exceeds the boundary $[a_j, b_j]$, it may not exceed the boundary of the whole search space. Therefore, when k is small in the current period, the imaging will exceed the boundary of the current j-th dimension, which helps to expand the search range, reduces the possibility of premature stagnation in the early stage, and helps to jump out of the local optimum. Finally, the greedy strategy is adopted. If the fitness value of the reverse solution is small and better than the original solution, the solution is updated and applied to the algorithm as follows:

$$x_{i,j}^{t+1}=\begin{cases} \dfrac{a_j^t+b_j^t}{2}+\dfrac{a_j^t+b_j^t}{2k}-\dfrac{x_{i,j}^{t+1}}{k}, & f\left(x^*\right)<f(x),\\ x_{i,j}^{t+1}, & \text{otherwise.} \end{cases} \tag{19}$$

3.3.1. Verify the Ability to Jump Out of Local Optimization. In this paper, the Shekel function mentioned above is selected as an example to draw the individual distribution diagram between the improved algorithm and the original algorithm to verify the ability of the above strategy to jump out of local optimization. The function image is shown in Figure 2. The selected population is 100, and the maximum number of iterations is 20. The individual distribution of the two algorithms are shown in Figures 5 and 6.

As can be seen from Figure 5, most individuals in the original algorithm have local aggregation and fall into local optimization. As can be seen from Figure 6, the improved

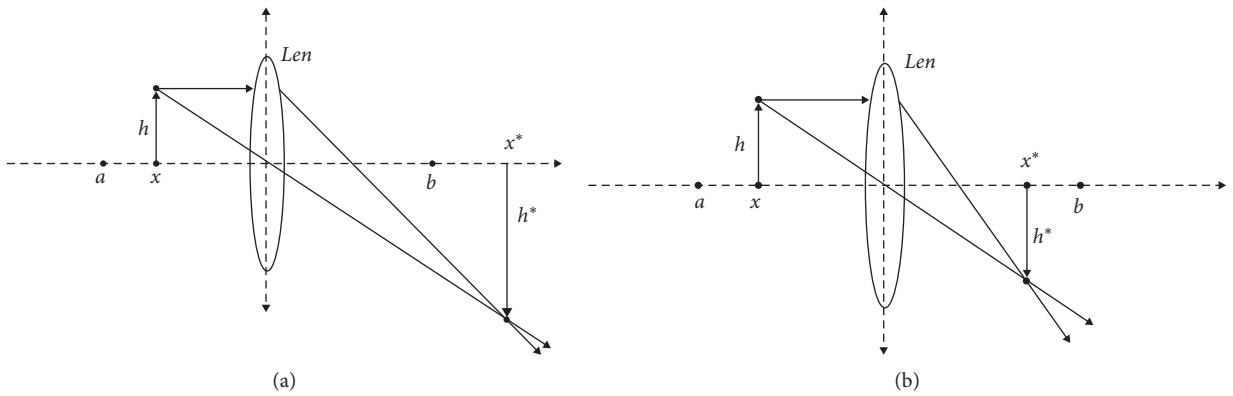


Figure 4: Main principles of lens imaging.

Table 2: Comparison of three kinds of reverse learning.

	Boundary	Focusing ability (k)	Reverse solution position	Effect
Reverse learning	Unchanged	1	On boundary midpoint symmetry	Accelerating convergence
Lens imaging learning	Dynamic change according to the maximum and minimum of individual position of population	A large constant, which is more than 1	About boundary midpoint reduction imaging	Accelerating convergence
Dimension by dimension lens imaging learning	The maximum and minimum values of each dimension change dynamically according to the individual position of the population	According to the dynamic change of iteration times, the first and middle stages are less than 1, and the later stage is more than 1	In the early and middle stages, the image is enlarged at the midpoint of the boundary, and in the later stage, the image is smaller	Jump out of the local optimum in the first and middle stage and accelerate the convergence in the later stage

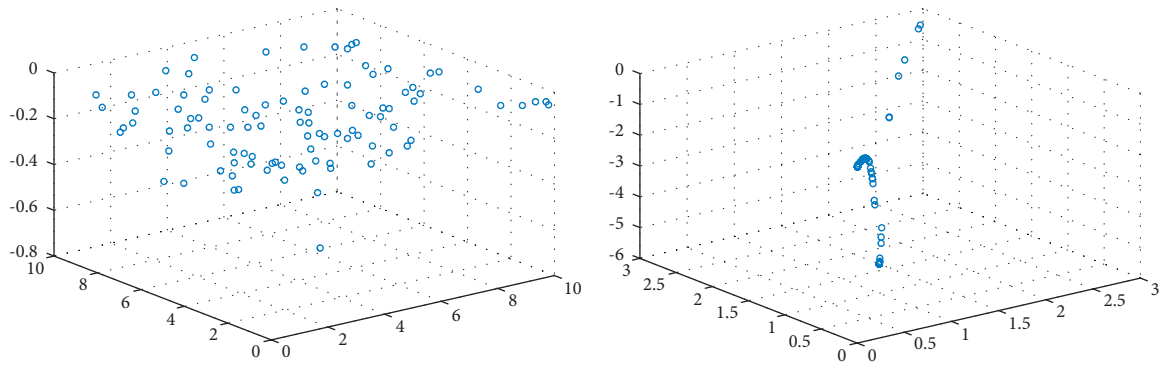


Figure 5: Individual distribution of SSA. (a) SSA individual initialization map. (b) Individual distribution of SSA in 20 generations.

algorithm has a larger search space, and most individuals converge near the optimal solution, effectively jumping out of the local optimal solution.

3.3.2. Proof of Convergence of Dimension by Dimension Lens Imaging Learning. The proof of convergence of general refraction reverse learning swarm intelligence algorithm is given in reference [37]. Here, its conclusion is introduced to prove the convergence of the ISSA for dimensional lens back learning. It should be pointed out that the proof of convergence does not necessarily ensure that the algorithm converges to the global optimal solution. Since the SSA is also a swarm intelligence search algorithm, there are the following theorems:

Theorem 1. *If the SSA algorithm based on general back learning converges, the ISSA algorithm also converges.*

Proof. Let $x_i(t)$ and $x_i^*(t)$ be the current solution and reverse solution of generation t, $x_{i,j}(t)$ and $x_{i,j}^*(t)$ be the values of $x_i(t)$ and $x_i^*(t)$ in the j-th dimension, respectively, and the global optimal solution is x^{best}. From Theorem 1,

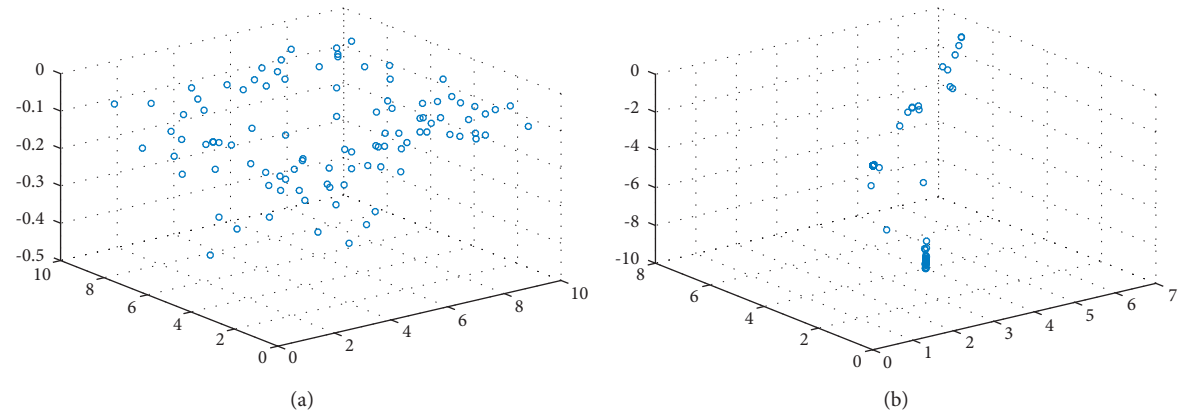


Figure 6: Individual distribution of ISSA. (a) ISSA individual initialization map. (b) Individual distribution of ISSA in 20 generations.

$$\lim_{t\longrightarrow\infty} x_i(t) = x^{\text{best}}. \tag{20}$$

Extended to the j-th dimension:

$$\lim_{t\longrightarrow\infty} x_{i,j}(t) = x_j^{\text{best}}. \tag{21}$$

Due to $a_j(t) = \min(x_j(t))$, $b_j(t) = \max(x_j(t))$, we get

$$\lim_{t\longrightarrow\infty} a_j(t) = \lim_{t\longrightarrow\infty} b_j(t) = x_j^{\text{best}}. \tag{22}$$

In the t-th generation, the current solution generated by the reverse learning strategy based on the lens imaging principle is

$$x_{i,j}^*(t) = \frac{a_j(t) + b_j(t)}{2} + \frac{a_j(t) + b_j(t)}{2k} - \frac{x_{i,j}(t)}{k}. \tag{23}$$

When $t \longrightarrow \infty$, from the above formula, we get

$$\begin{aligned}\lim_{t\longrightarrow\infty} x_{i,j}^*(t) &= \lim_{t\longrightarrow\infty} \frac{a_j(t) + b_j(t)}{2} + \frac{a_j(t) + b_j(t)}{2k} - \frac{x_{i,j}(t)}{k} \\ &= \lim_{t\longrightarrow\infty} \frac{x_j^{\text{best}}(t) + x_j^{\text{best}}(t)}{2} + \frac{x_j^{\text{best}}(t) + x_j^{\text{best}}(t)}{2k} \\ &\quad - \frac{x_j^{\text{best}}}{k} = x_j^{\text{best}}.\end{aligned} \tag{24}$$

Return to the whole dimension:

$$\lim_{t\longrightarrow\infty} x_i^*(t) = x^{\text{best}}. \tag{25}$$

It can be seen that when $x_i(t)$ converges to x^{best}, the inverse solution $x_i^*(t)$ generated by lens learning also converges to x^{best}. Therefore, if the SSA based on general reverse learning converges, the ISSA also converges. □

3.4. Improved Boundary Control. In the standard SSA, when an individual in the optimization process exceeds the boundary, the boundary control will be carried out for the individual. The principle is as follows:

$$x_{i,j}^{t+1} = \begin{cases} \text{Ub}, & x_{i,j}^{t+1} > \text{Ub}, \\ \text{Lb}, & x_{i,j}^{t+1} < \text{Lb}. \end{cases} \tag{26}$$

Ub and Lb are the upper and lower bounds of space, respectively. In this method, the strategy of turning the individuals beyond the boundary into the boundary will lead to the aggregation of individuals at the boundary and reduce the diversity of the population. It can be seen from Table 1 that although the improved strategy reduces the number of individuals beyond the boundary, some individuals still exceed the boundary and gather at the boundary. Literature [38] adopts the strategy of the current optimal solution for individuals beyond the boundary, which will be difficult to get rid of the local optimization. Therefore, this paper proposes a simple boundary treatment method, namely,

$$x_{i,j}^{t+1} = \text{Lb} + (\text{Ub} - \text{Lb}) \cdot \text{rand}(), \quad x_{i,j}^{t+1} > \text{Ub} \text{ and } x_{i,j}^{t+1} < \text{Lb}. \tag{27}$$

In this method, the individuals beyond the boundary are randomly assigned to the search space, which makes more effective use of the population individuals and increases the diversity of the population more than that of the original algorithm.

3.5. Improved Sparrow Search Algorithm Flow. In this paper, an improved sparrow search algorithm based on iterative local search is proposed. Firstly, the variable helix factor is used to improve the extensive search of followers, which reduces the individuals beyond the boundary and speeds up the convergence speed in the later stage. Secondly, the improved iterative local search is used to improve the local search of the followers. The initial solution is subject to local search and iterative local search after disturbance, which makes full use of the current position information to prevent premature convergence and improve the quality and accuracy of understanding. Then, the lens with changed focusing ability is used to carry out dimension by dimension lens imaging learning for the scouter, and it increases the search space and helps the population jump out of the local optimum. Finally, the boundary control strategy is improved

to reduce the aggregation of individuals at the boundary and increase the diversity of the population. The introduction of various strategies makes the algorithm more flexible in the optimization ability, makes the population more diverse, strengthens its ability to get rid of local optimization, balances the global search ability and local search ability of the algorithm, and is conducive to finding reliable solutions. The specific algorithm flow is shown in Figure 7. The specific pseudo code is as follows:

3.6. Time Complexity Analysis. Time complexity is an important index to measure the performance of the algorithm, which is used to measure the running time of the algorithm. Assuming that the population size of the algorithm is P, the maximum number of iterations is M and the dimension is D, and the time complexity of the sparrow search algorithm is $O(P{\cdot}M{\cdot}D)$. From a macro point of view, the improved sparrow search algorithm does not change the structure and cycle times of the algorithm, so its time complexity is also $O(P{\cdot}M{\cdot}D)$, which is consistent with the original algorithm. From the microscopic point of view, the greedy strategy is adopted for iterative local search and dimension by dimension lens learning, which increases the algorithm complexity of some followers and all scouters to a certain extent, but the introduction of the improved strategy does not improve the order of magnitude of the algorithm, so the time complexity is still $O(P{\cdot}M{\cdot}D)$.

4. Benchmark Function Test

In order to better verify the performance of ISSA, this paper selects 23 common basic test functions for verification and tests and compares them with 10 algorithms including PSO, SCA, GWO, WOA, MWOA, SSA, BSSA, CSSA, and LSSA. The specific parameter settings are shown in Table 3, and the test function information is shown in Table 4. F1–F7 is the high-dimensional single-peak benchmark function, F8–F13 is the high-dimensional multipeak benchmark function, and F14–F23 is the low-dimensional multipeak benchmark function. F1–F13 is tested in 100 dimensions to verify the performance of the algorithm in higher dimensions. For the sake of fairness, the population size and maximum number of iterations of each algorithm are 30 and 500, respectively, and each algorithm is run independently for 30 times to calculate its best value (best), worst value (worst), average value (AVE), and standard deviation (STD), and the optimal value of each index is processed in bold. Finally, each algorithm will be ranked according to the average value of the algorithm in the function. When the average values of the two algorithms are equal, the standard deviation will be compared. For performance evaluation, simulations are performed on Windows 10 Matlab 2016a, AMD Ryzen 7 4800U with Radeon Graphics @1.80 GHz with 16 GB RAM.

ISSA ranks first in most functions, and its average ranking is better than that of other algorithms. It can be seen from Tables 5 and 6 that, compared with other algorithms, ISSA has found the theoretical optimal value except F7, F10, F12, F13, and F15 and has found the optimal value in all

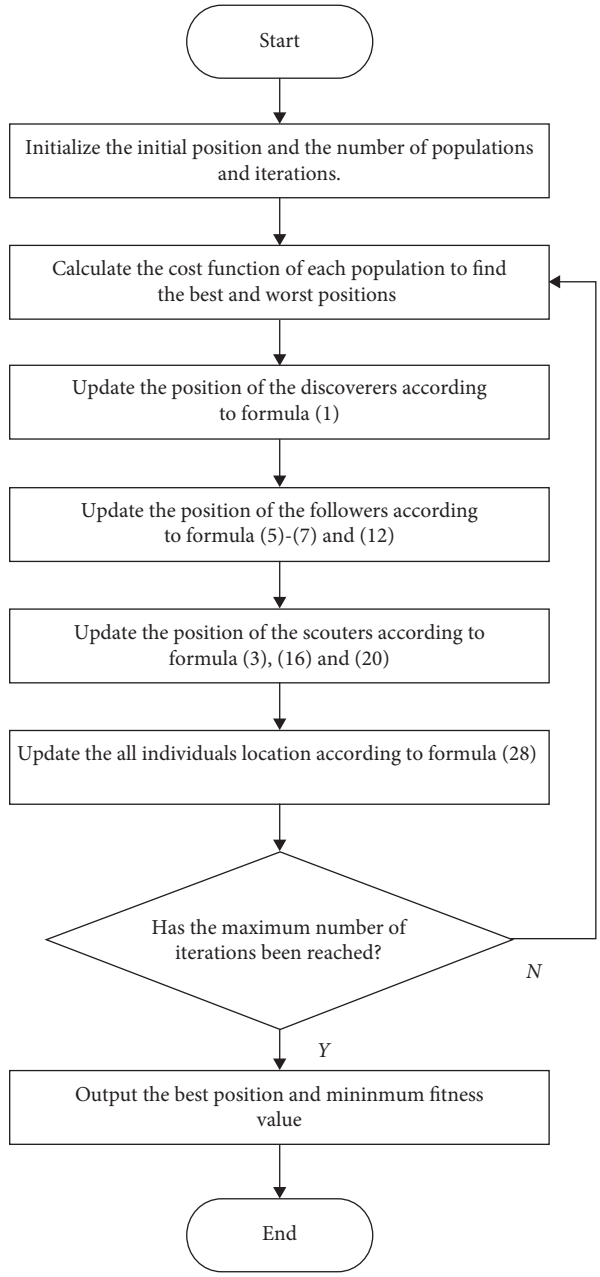


Figure 7: Algorithm flow chart.

algorithms in the above five functions. It can be seen that the ISSA has good ability to find the optimal solution. Among them, only the ISSA in F5 finds the theoretical optimal value, and only SSA and ISSA in F6 find the optimal value. The optimization accuracy of ISSA in F12 and F13 is a large number of orders of magnitude higher than that of other algorithms. When the dimension increases to 100, the optimization performance of ISSA is still very stable, while PSO, SCA, WOA, GOA, and SSA have a great impact on the optimization ability. It is worth mentioning that in F6 and F12, ISSA is only better than SSA and SSA improved the algorithm in terms of optimal value, but other indicators are better than other swarm intelligence algorithms. In F1–F13 function, ISSA finds a better solution than or the same

Input:
M: maximum number of iterations
PD: proportion of discoverer
SD: proportion of scouters
R_2: alert value
N: population sparrows
Output: X_{best}, f_g
Initialize population
$t = 1$;
While ($t < M$)
  Find the position of the best and worst sparrow individuals according to fitness values.
  $R_2 = \text{rand}\ (1)$
  For $i = 1 : \text{PD}$
  Update the location of the discoverers according to formula (1);
  End for
  For $i = (\text{PD} + 1) : \text{N}$
  Update the location of the followers according to formulas (5)–(7) and (12);
  End for
  For $l = 1 : \text{SD}$
  Get the individual position of a sparrow that is aware of danger according to formulas (3), (16), and (19);
  End for
Get the location of the new optimal individual;
Update the all individuals' location according to formula (27);
$t = t + 1$
End while
Return: X_{best}, f_g

Algorithm 1: The framework of the ISSA.

Table 3: Parameter.

Algorithm	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
Parameter	$c1 = 2$ $c2 = 2$ Wmix = 0.2 Wmax = 0.9	$a = 2$	$a = (2 \longrightarrow 0)$	$b = 1$	$b = 1$ $n = 12000$	ST = 0.8 PD = 0.2 SD = 0.2	ST = 0.8 PD = 0.2 SD = 0.2	ST = 0.8 PD = 0.2 SD = 0.2	ST = 0.8 PD = 0.2 SD = 0.2	ST = 0.8 PD = 0.2 SD = 0.2 $a = 0.1$

solution as SSA, indicating that ISSA does not reduce the optimization ability of the algorithm. In the fixed dimension F14–F23, ISSA algorithm has good stability and the ability to jump out of the local optimum. It can find the solution close to the theoretical optimum almost every time.

In order to better describe the optimization ability, difference, and convergence speed of the algorithm with other algorithms, the Wilcoxon rank sum test results with other algorithms is given in Table 7 according to ISSA, and the convergence diagram of each algorithm is given in Figure 8. At the same time, in order to test the contribution of the four strategies to the algorithm, this paper selects a basic function to test the four components of the improved algorithm, as shown in the first figure in Figure 8. The above is based on the test results of 23 benchmark functions.

In this paper, the four strategies and the original SSA and ISSA are tested in the F1 function of the basic test function, as shown in the first figure in Figure 8. ISSA1, ISSA2, ISSA3, and ISSA4 represent the algorithms improved separately by the four improved strategies proposed before. Among them, ISSA2, ISSA3, and ISSA have found the theoretical optimal value. Compared with the original SSA, ISSA2 has significantly improved the convergence speed, reflecting its excellent search ability; ISSA3 suddenly converges to the theoretical optimal value in the middle of the iteration, indicating its excellent ability to jump out of the local optimal. Compared with the original SSA algorithm, the optimization results and convergence speed of ISSA1 and ISSA4 are improved by dozens of orders of magnitude, which can improve to a certain extent. It can be seen that improved iterative local search and dimension by dimension lens imaging learning play a more critical role in improving the ability of the algorithm. The combination of the above four strategies enables us to obtain an ISSA with faster convergence speed, more accurate results, and more stability.

In the Wilcoxon rank sum test, when the value is less than 0.05, it can be considered that there is a significant difference between the two. In Table 7, NaN indicates that their performance is equivalent and cannot be compared. It can be seen that most values are less than 0.05, indicating that the optimization performance of ISSA is significantly different from other algorithms, among which the difference between ISSA and CSSA is the

Table 4: Test function.

Function	Dimensions	Interval	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/100	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30/100	$[-100, 100]$	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30/100	$[-100, 100]$	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30/100	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30/100	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30/100	$[-100, 100]$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30/100	$[-1.28, 1.28]$	0
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30/100	$[-500, 500]$	$-418.98n$
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30/100	$[-5.12, 5.12]$	0
$F_{10}(x) = -20\exp(-0.2\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}) - \exp[(1/n)\sum_{i=1}^{n} \cos(2\pi x_i)] + 20 + e$	30/100	$[-32, 32]$	0
$F_{11}(x) = (1/4000)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	30/100	$[-600, 600]$	0
$F_{12}(x) = (\pi/n)\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30/100	$[-50, 50]$	0
$y_i = 1 + ((x_i + 1)/4)$			
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30/100	$[-50, 50]$	0
$F_{14}(x) = ((1/500) + \sum_{j=1}^{25} 1/(j + \sum_{i=1}^{2} (x_i - a_{ij})^6))^{-1}$	2	$[-65.536, 65.536]$	0.998
$F_{15}(x) = \sum_{i=1}^{11} (a_i - (x_1(b_i^2 + b_1 x_2)/(b_i^2 + b_1 x_3 + x_4)))^2$	4	$[-5, 5]$	0.0003
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.032
$F_{17}(x) = (x_2 - (5.1/4\pi^2)x_1 - 6)^2 + 10(1 - (1/8\pi))\cos x_1 + 10$	2	$[-5, 5]$	0.3979
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
$F_{19}(x) = \sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	3	$[0, 1]$	-3.863
$F_{20}(x) = \sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	6	$[0, 1]$	-3.32
$F_{21}(x) = \sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532
$F_{22}(x) = \sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4029
$F_{23}(x) = \sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5364

Table 5: Comparison table of the optimization effect of each algorithm (30 dimensions and fixed dimensions).

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
	Best	$3.039E-08$	$2.140E-04$	$5.309E-40$	$3.217E-106$	$2.455E-106$	**0**	$5.308E-269$	**0**	**0**	**0**
	Worst	$2.030E-06$	$1.076E+01$	$9.250E-38$	$3.331E-94$	$1.392E-95$	**0**	$9.853E-249$	**0**	**0**	**0**
F1	Ave	$2.356E-07$	$8.143E-01$	$1.680E-38$	$1.167E-95$	$1.526E-96$	**0**	$3.337E-250$	**0**	**0**	**0**
	Std	$3.697E-07$	$2.128E+00$	$2.379E-38$	$6.075E-95$	$3.710E-96$	**0**	**0**	**0**	**0**	**0**
	Rank	9	10	8	7	6	1	5	1	1	1
	Best	$5.760E-05$	$4.632E-05$	$3.354E-23$	$2.808E-63$	$1.441E-62$	**0**	$1.827E-136$	**0**	$2.123E-263$	**0**
	Worst	$2.620E-03$	$1.803E-02$	$5.443E-22$	$7.961E-57$	$3.663E-55$	$5.557E-231$	$1.955E-126$	**0**	$2.174E-169$	**0**
F2	Ave	$4.306E-04$	$4.924E-03$	$1.410E-22$	$1.014E-57$	$1.594E-56$	$2.250E-232$	$1.332E-127$	**0**	$7.253E-171$	**0**
	Std	$4.897E-04$	$4.613E-03$	$1.008E-22$	$2.073E-57$	$6.709E-56$	**0**	$4.462E-127$	**0**	**0**	**0**
	Rank	9	10	8	6	7	3	5	1	4	1
	Best	$4.148E+01$	$7.288E+02$	$1.144E-07$	$1.056E+04$	$2.370E+04$	**0**	$6.239E-263$	**0**	**0**	**0**
	Worst	$1.465E+02$	$1.962E+04$	$3.338E-03$	$5.602E+04$	$6.525E+04$	$6.949E-59$	$3.651E-232$	**0**	$1.751E-244$	**0**
F3	Ave	$9.521E+01$	$7.423E+03$	$1.535E-04$	$3.870E+04$	$4.355E+04$	$2.316E-60$	$1.217E-233$	**0**	$7.297E-246$	**0**
	Std	$3.092E+01$	$4.667E+03$	$6.126E-04$	$1.181E+04$	$1.007E+04$	$1.269E-59$	**0**	**0**	**0**	**0**
	Rank	7	8	6	9	10	5	4	1	3	1
	Best	$6.129E-01$	$1.282E+01$	$9.960E-08$	$3.426E+00$	$4.515E-02$	$2.102E-293$	$3.783E-130$	**0**	$1.942E-210$	**0**
	Worst	$1.677E+00$	$6.014E+01$	$5.466E-06$	$8.940E+01$	$8.833E+01$	$8.277E-36$	$6.088E-120$	$1.338E-182$	$1.789E-142$	**0**
F4	Ave	$1.077E+00$	$3.790E+01$	$1.135E-06$	$4.785E+01$	$4.205E+01$	$2.774E-37$	$4.300E-121$	$4.720E-184$	$6.026E-144$	**0**
	Std	$2.676E-01$	$1.281E+01$	$1.271E-06$	$2.867E+01$	$3.029E+01$	$1.511E-36$	$1.332E-120$	**0**	$3.265E-143$	**0**
	Rank	7	8	6	10	9	5	4	2	3	1
	Best	$2.564E+01$	$4.001E+01$	$2.581E+01$	$2.707E+01$	$2.701E+01$	$2.124E-14$	$2.502E-08$	$1.176E-09$	$8.738E-10$	**0**
	Worst	$1.817E+02$	$2.485E+05$	$2.874E+01$	$2.877E+01$	$2.876E+01$	$3.498E-05$	$2.572E-04$	$5.266E-04$	$1.097E-03$	**0**
F5	Ave	$8.783E+01$	$2.463E+04$	$2.704E+01$	$2.794E+01$	$2.795E+01$	$4.175E-06$	$5.466E-05$	$2.864E-05$	$5.689E-05$	**0**
	Std	$4.634E+01$	$4.827E+04$	$6.734E-01$	$4.754E-01$	$4.112E-01$	$9.187E-06$	$7.765E-05$	$1.005E-04$	$2.064E-04$	**0**
	Rank	9	10	6	7	8	2	4	3	5	1
	Best	$7.636E-06$	$5.178E+00$	$6.012E-07$	$4.835E-02$	$6.880E-02$	**0**	$3.524E-11$	$1.360E-12$	$3.385E-10$	**0**
	Worst	$1.739E-03$	$5.039E+01$	$1.255E+00$	$9.709E-01$	$1.047E+00$	$7.534E-06$	$1.103E-07$	**5.188E−08**	$3.010E-07$	$7.059E-04$
F6	Ave	$2.531E-04$	$1.420E+01$	$7.218E-01$	$4.287E-01$	$3.361E-01$	$3.019E-07$	$1.788E-08$	**6.178E−09**	$3.970E-08$	$1.409E-04$
	Std	$3.327E-04$	$1.305E+01$	$3.225E-01$	$2.176E-01$	$2.303E-01$	$1.372E-06$	$2.881E-08$	**1.170E−08**	$7.458E-08$	$1.965E-04$
	Rank	6	10	9	8	7	4	2	1	3	5
	Best	$7.982E-02$	$1.117E-02$	$3.244E-04$	$5.252E-05$	$5.025E-05$	$1.287E-05$	$2.264E-05$	$1.056E-05$	$1.083E-05$	**8.231E−06**
	Worst	$3.788E-01$	$2.485E-01$	$4.318E-03$	$1.108E-02$	$1.460E-02$	$1.062E-03$	$9.727E-04$	**4.982E−04**	$8.427E-04$	$5.115E-04$
F7	Ave	$1.778E-01$	$8.566E-02$	$1.795E-03$	$2.821E-03$	$3.241E-03$	$2.440E-04$	$2.722E-04$	$1.632E-04$	$2.355E-04$	**1.506E−04**
	Std	$6.799E-02$	$6.210E-02$	$1.129E-03$	$2.905E-03$	$3.662E-03$	$2.364E-04$	$2.438E-04$	$1.211E-04$	$1.908E-04$	**1.196E−04**
	Rank	10	9	6	7	8	4	5	2	3	1
	Best	$-8.128E+03$	$-4.963E+03$	$-7.948E+03$	$-1.257E+04$	$-1.257E+04$	**−1.257E+04**	$-1.030E+04$	$-1.257E+04$	$-1.257E+04$	**−1.257E+04**
	Worst	$-3.967E+03$	$-3.578E+03$	$-3.642E+03$	$-8.304E+03$	$-7.837E+03$	$-7.673E+03$	$-7.671E+03$	$-8.952E+03$	$-1.033E+04$	**−1.083E+04**
F8	Ave	$-6.667E+03$	$-4.023E+03$	$-6.361E+03$	$-1.153E+04$	$-1.142E+04$	$-1.060E+04$	$-8.891E+03$	$-1.128E+04$	$-1.182E+04$	**−1.245E+04**
	Std	$9.060E+02$	$3.243E+02$	$7.713E+02$	$1.520E+03$	$1.230E+03$	$1.948E+03$	$5.289E+02$	$8.299E+02$	$6.529E+02$	**3.565E+02**
	Rank	8	10	9	3	4	6	7	5	2	1

Table 5: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
	Best	$2.609E+01$	$8.825E-01$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Worst	$8.661E+01$	$2.056E+02$	$1.045E+01$	**0**	$1.137E-13$	**0**	**0**	**0**	**0**	**0**
F9	Ave	$5.757E+01$	$3.868E+01$	$1.057E+00$	**0**	$3.790E-15$	**0**	**0**	**0**	**0**	**0**
	Std	$1.410E+01$	$4.056E+01$	$2.499E+00$	**0**	$2.076E-14$	**0**	**0**	**0**	**0**	**0**
	Rank	10	9	8	1	7	1	1	1	1	1
	Best	$2.946E-03$	$2.093E-02$	$7.905E-14$	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**
	Worst	$1.341E+00$	$2.035E+01$	$2.780E-13$	$7.994E-15$	$7.994E-15$	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**
F10	Ave	$2.508E-01$	$1.104E+01$	$1.554E-13$	$3.849E-15$	$5.033E-15$	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**	**8.882E − 16**
	Std	$4.770E-01$	$9.347E+00$	$4.636E-14$	$2.483E-15$	$2.653E-15$	**0**	**0**	**0**	**0**	**0**
	Rank	9	10	8	6	7	1	1	1	1	1
	Best	$1.262E-06$	$4.291E-01$	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Worst	$3.694E-02$	$2.192E+00$	$2.243E-02$	$2.284E-01$	**0**	**0**	**0**	**0**	**0**	**0**
F11	Ave	$9.120E-03$	$1.013E+00$	$4.558E-03$	$7.613E-03$	**0**	**0**	**0**	**0**	**0**	**0**
	Std	$1.036E-02$	$2.950E-01$	$7.488E-03$	$4.170E-02$	**0**	**0**	**0**	**0**	**0**	**0**
	Rank	9	10	7	8	1	1	1	1	1	1
	Best	$4.722E-08$	$1.373E+00$	$6.907E-03$	$4.668E-03$	$2.905E-03$	$5.172E-19$	$3.481E-13$	$4.456E-13$	$2.676E-09$	**1.571E − 32**
	Worst	$1.037E-01$	$3.061E+06$	$1.258E-01$	$3.701E-01$	$2.246E-01$	$1.549E-07$	$8.831E-09$	**3.088E − 09**	$6.638E-07$	$9.777E-05$
F12	Ave	$6.914E-03$	$1.092E+05$	$4.088E-02$	$3.888E-02$	$2.754E-02$	$7.364E-09$	$8.349E-10$	**2.263E − 10**	$1.373E-07$	$2.558E-05$
	Std	$2.630E-02$	$5.587E+05$	$2.534E-02$	$6.678E-02$	$3.888E-02$	$2.905E-08$	$1.843E-09$	**5.697E − 10**	$1.673E-07$	$3.011E-05$
	Rank	6	10	9	8	7	3	2	1	4	5
	Best	$4.356E-06$	$5.547E+00$	$1.045E-01$	$2.076E-01$	$1.165E-01$	$1.010E-21$	$1.163E-11$	$2.611E-16$	$1.351E-15$	**1.350E − 32**
	Worst	$1.152E-02$	$3.410E+06$	$9.183E-01$	$1.069E+00$	$1.303E+00$	$2.333E-07$	$2.900E-07$	$1.083E-10$	$3.592E-11$	**1.350E − 32**
F13	Ave	$5.321E-03$	$2.239E+05$	$5.196E-01$	$6.395E-01$	$5.154E-01$	$2.072E-08$	$2.126E-08$	$1.612E-11$	$3.754E-12$	**1.350E − 32**
	Std	$5.498E-03$	$7.069E+05$	$1.948E-01$	$2.354E-01$	$2.762E-01$	$5.998E-08$	$6.089E-08$	$2.840E-11$	$8.284E-12$	**0**
	Rank	6	10	8	9	7	4	5	3	2	1
	Best	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**	**9.980E − 01**
	Worst	$1.172E+01$	$1.076E+01$	$1.267E+01$	$1.076E+01$	$1.076E+01$	$1.267E+01$	$1.267E+01$	**9.980E − 01**	$1.267E+01$	$1.267E+01$
F14	Ave	$4.147E+00$	$2.251E+00$	$4.065E+00$	$2.767E+00$	$3.907E+00$	$8.622E+00$	$1.086E+01$	**9.980E − 01**	$1.585E+00$	$1.453E+00$
	Std	$3.028E+00$	$1.888E+00$	$4.163E+00$	$2.953E+00$	$3.791E+00$	$5.438E+00$	$4.143E+00$	**4.517E − 16**	$2.179E+00$	$2.149E+00$
	Rank	8	4	7	5	6	9	10	1	3	2
	Best	$3.084E-04$	$4.873E-04$	$3.075E-04$	$3.081E-04$	$3.217E-04$	**3.075E − 04**	**3.075E − 04**	**3.075E − 04**	**3.075E − 04**	**3.075E − 04**
	Worst	$1.093E-03$	$1.506E-03$	$2.036E-02$	$5.941E-03$	$2.252E-03$	$3.447E-04$	$6.256E-04$	$3.205E-04$	**3.155E − 04**	$3.319E-04$
F15	Ave	$8.461E-04$	$1.002E-03$	$4.462E-03$	$1.216E-03$	$6.588E-04$	$3.098E-04$	$3.204E-04$	**3.079E − 04**	$3.080E-04$	$3.100E-04$
	Std	$1.588E-04$	$3.287E-04$	$8.091E-03$	$1.271E-03$	$4.499E-04$	$7.828E-06$	$5.824E-05$	$2.380E-06$	**1.552E − 06**	$6.408E-06$
	Rank	7	8	10	9	6	3	5	1	2	4
	Best	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**
	Worst	**−1.032E + 00**	**−1.031E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	$-2.155E-01$	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**
F16	Ave	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	$-1.004E+00$	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**	**−1.032E + 00**
	Std	**0**	$7.279E-05$	**0**	**0**	**0**	$1.490E-01$	**0**	**0**	**0**	**0**
	Rank	1	9	1	1	1	10	1	1	1	1

Table 5: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
	Best	$3.979E-01$	$3.979E-01$	$3.979E-01$	$3.979E-01$	$3.979E-01$	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**
	Worst	$3.979E-01$	$4.108E-01$	$3.985E-01$	$3.980E-01$	$3.980E-01$	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**
F17	Ave	**3.979E − 01**	$3.994E-01$	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**	**3.979E − 01**
	Std	**0**	$2.363E-03$	$1.095E-04$	$1.569E-05$	$2.801E-05$	**0**	**0**	**0**	**0**	**0**
	Rank	1	10	9	7	8	1	1	1	1	1
	Best	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**
	Worst	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	$3.000E+01$	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**
F18	Ave	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	$4.800E+00$	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**	**3.000E + 00**
	Std	**0**	**0**	**0**	**0**	**0**	$6.850E+00$	**0**	**0**	**0**	**0**
	Rank	1	1	1	1	1	10	1	1	1	1
	Best	**−3.863E + 00**	$-3.862E+00$	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**
	Worst	**−3.863E + 00**	$-3.854E+00$	$-3.856E+00$	$-3.861E+00$	$-3.860E+00$	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**
F19	Ave	**−3.863E + 00**	$-3.855E+00$	$-3.862E+00$	$-3.862E+00$	$-3.862E+00$	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**	**−3.863E + 00**
	Std	$3.162E-15$	$1.785E-03$	$1.528E-03$	$6.637E-04$	$6.423E-04$	**0**	**0**	**0**	**0**	**0**
	Rank	6	10	7	8	9	1	1	1	1	1
	Best	**−3.322E + 00**	$-3.224E+00$	**−3.322E + 00**	**−3.322E + 00**	**−3.322E + 00**	**−3.322E + 00**	**−3.322E + 00**	**−3.322E + 00**	**−3.322E + 00**	**−3.322E + 00**
	Worst	**−3.203E + 00**	$-2.991E+00$	$-3.138E+00$	$-3.082E+00$	$-3.087E+00$	**−3.203E + 00**	**−3.203E + 00**	**−3.203E + 00**	**−3.203E + 00**	**−3.203E + 00**
F20	Ave	$-3.270E+00$	$-3.078E+00$	$-3.247E+00$	$-3.258E+00$	$-3.243E+00$	$-3.263E+00$	$-3.286E+00$	$-3.263E+00$	$-3.247E+00$	**−3.318E + 00**
	Std	$5.993E-02$	$6.050E-02$	$6.344E-02$	$7.915E-02$	$6.898E-02$	$6.047E-02$	$5.542E-02$	$6.047E-02$	$5.828E-02$	**2.171E − 02**
	Rank	3	10	7	6	9	4	2	5	8	1
	Best	$-1.015E+01$	$-5.819E+00$	$-1.015E+01$	$-1.015E+01$	$-1.015E+01$	$-1.015E+01$	**−1.015E + 01**	**−1.015E + 01**	**−1.015E + 01**	**−1.015E + 01**
	Worst	$-2.631E+00$	$-4.973E-01$	$-2.683E+00$	$-4.984E+00$	$-2.615E+00$	$-5.055E+00$	$-9.996E+00$	**−1.015E + 01**	**−1.015E + 01**	**−1.015E + 01**
F21	Ave	$-7.630E+00$	$-2.164E+00$	$-9.088E+00$	$-8.265E+00$	$-8.515E+00$	$-9.303E+00$	$-1.013E+01$	**−1.015E + 01**	**−1.015E + 01**	**−1.015E + 01**
	Std	$3.218E+00$	$1.734E+00$	$2.466E+00$	$2.490E+00$	$2.753E+00$	$1.932E+00$	$3.978E-02$	**0**	**0**	**0**
	Rank	9	10	6	8	7	5	4	1	1	1
	Best	**−6.207E + 00**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**	**−1.040E + 01**
	Worst	$-9.028E-01$	$-5.129E+00$	$-1.838E+00$	$-1.835E+00$	$-5.088E+00$	$-5.088E+00$	**−1.040E + 01**	**−1.040E + 01**	$-5.088E+00$	**−1.040E + 01**
F22	Ave	$-2.986E+00$	$-1.023E+01$	$-7.664E+00$	$-7.782E+00$	$-8.985E+00$	$-9.850E+00$	**−1.040E + 01**	**−1.040E + 01**	$-9.340E+00$	**−1.040E + 01**
	Std	$1.774E+00$	$9.629E-01$	$3.007E+00$	$3.125E+00$	$2.391E+00$	$1.615E+00$	**0**	**0**	$2.162E+00$	**0**
	Rank	10	4	9	8	7	5	1	1	6	1
	Best	**−1.054E + 01**	$-8.411E+00$	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**
	Worst	$-2.422E+00$	$-9.439E-01$	$-5.129E+00$	$-2.422E+00$	$-2.803E+00$	$-5.129E+00$	$-1.026E+01$	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**
F23	Ave	$-9.395E+00$	$-3.822E+00$	$-1.018E+01$	$-7.221E+00$	$-8.252E+00$	$-9.815E+00$	$-1.051E+01$	**−1.054E + 01**	**−1.054E + 01**	**−1.054E + 01**
	Std	$2.644E+00$	$1.569E+00$	$1.366E+00$	$3.235E+00$	$2.843E+00$	$1.870E+00$	$7.007E-02$	**0**	**0**	**0**
	Rank	7	10	5	9	8	6	4	1	1	1
Average rank	4.60	5.93	4.76	4.45	4.38	2.93	2.14	1.17	1.72	1.10	

Table 6: Comparison table of the optimization effect of each algorithm (100 dimensions).

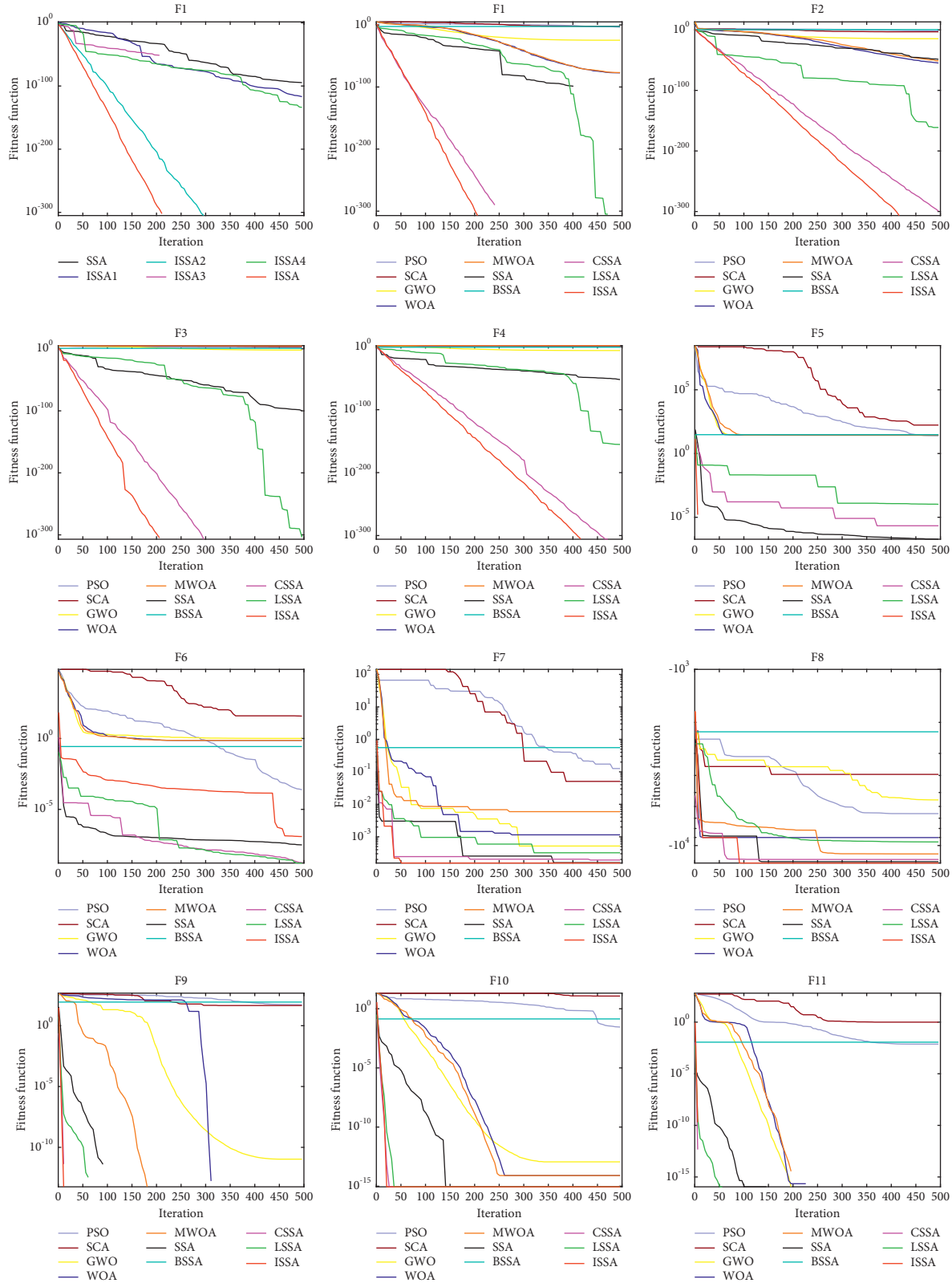
F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
	Best	$9.872E+00$	$9.308E+02$	$2.693E-12$	$4.175E-83$	$2.894E-84$	**0**	$2.714E-255$	**0**	**0**	**0**
	Worst	$3.569E+01$	$2.981E+04$	$1.824E-10$	$8.142E-70$	$4.168E-70$	$2.928E-66$	$9.366E-235$	**0**	$1.063E-272$	**0**
F1	Ave	$2.206E+01$	$1.131E+04$	$4.086E-11$	$4.068E-71$	$1.464E-71$	$1.010E-67$	$3.248E-236$	**0**	$3.935E-274$	**0**
	Std	$5.407E+00$	$7.239E+03$	$4.038E-11$	$1.574E-70$	$7.601E-71$	$5.438E-67$	**0**	**0**	**0**	**0**
	Rank	9	10	8	6	5	7	4	1	3	1
	Best	$2.295E+01$	$1.120E-01$	$1.379E-07$	$4.706E-55$	$4.940E-57$	**0**	$1.226E-127$	**0**	$9.989E-215$	**0**
	Worst	$5.920E+01$	$2.205E+01$	$5.637E-07$	$3.020E-47$	$1.360E-48$	$8.218E-35$	$1.609E-115$	**0**	$4.528E-140$	**0**
F2	Ave	$3.901E+01$	$7.362E+00$	$2.368E-07$	$1.053E-48$	$1.017E-49$	$3.229E-36$	$5.982E-117$	**0**	$1.509E-141$	**0**
	Std	$9.152E+00$	$6.263E+00$	$8.618E-08$	$5.508E-48$	$3.266E-49$	$1.515E-35$	$2.945E-116$	**0**	$8.268E-141$	**0**
	Rank	10	9	8	6	5	7	4	1	3	1
	Best	$1.196E+04$	$1.540E+05$	$2.370E+01$	$4.152E+05$	$5.261E+05$	**0**	$9.631E-256$	**0**	**0**	**0**
	Worst	$2.610E+04$	$4.486E+05$	$3.307E+03$	$1.682E+06$	$1.491E+06$	$5.484E-51$	$1.912E-231$	**0**	**0**	**0**
F3	Ave	$1.660E+04$	$2.370E+05$	$1.037E+03$	$1.007E+06$	$1.040E+06$	$1.906E-52$	$6.898E-233$	**0**	**0**	**0**
	Std	$3.793E+03$	$6.543E+04$	$8.578E+02$	$3.146E+05$	$2.624E+05$	$1.018E-51$	**0**	**0**	**0**	**0**
	Rank	7	8	6	9	10	5	4	1	1	1
	Best	$9.191E+00$	$8.607E+01$	$7.311E-01$	$2.040E+01$	$1.568E+00$	$2.570E-282$	$1.681E-130$	**0**	$3.573E-261$	**0**
	Worst	$1.493E+01$	$9.584E+01$	$8.765E+00$	$9.697E+01$	$9.638E+01$	$7.474E-32$	$4.501E-118$	**0**	$4.421E-158$	**0**
F4	Ave	$1.220E+01$	$9.061E+01$	$2.931E+00$	$7.732E+01$	$7.614E+01$	$2.491E-33$	$2.897E-119$	**0**	$1.474E-159$	**0**
	Std	$1.550E+00$	$2.355E+00$	$1.931E+00$	$2.188E+01$	$2.359E+01$	$1.365E-32$	$9.208E-119$	**0**	0	**0**
	Rank	7	10	6	9	8	5	4	1	3	1
	Best	$7.371E+03$	$2.331E+07$	$9.657E+01$	$9.745E+01$	$9.752E+01$	$1.531E-14$	$1.382E-07$	$2.260E-08$	$8.400E-10$	**0**
	Worst	$3.114E+04$	$2.554E+08$	$9.849E+01$	$9.858E+01$	$9.846E+01$	$3.712E-03$	$2.409E-03$	$3.776E-04$	$1.007E-03$	**0**
F5	Ave	$1.571E+04$	$1.182E+08$	$9.778E+01$	$9.819E+01$	$9.814E+01$	$1.501E-04$	$3.859E-04$	$8.782E-05$	$2.173E-04$	**0**
	Std	$5.206E+03$	$6.138E+07$	$5.923E-01$	$2.454E-01$	$2.609E-01$	$6.750E-04$	$6.669E-04$	$1.081E-04$	$2.744E-04$	**0**
	Rank	9	10	6	8	7	3	5	2	4	1
	Best	$1.156E+01$	$2.016E+03$	$7.714E+00$	$2.371E+00$	$1.979E+00$	**0**	$7.437E-10$	$1.845E-09$	$8.530E-10$	**0**
	Worst	$3.269E+01$	$3.267E+04$	$1.146E+01$	$8.063E+00$	$6.289E+00$	**2.874E−06**	$1.995E-05$	$4.954E-06$	$1.287E-04$	$7.775E-03$
F6	Ave	$2.134E+01$	$1.089E+04$	$9.632E+00$	$4.301E+00$	$3.952E+00$	**1.845E−07**	$1.796E-06$	$6.391E-07$	$7.234E-06$	$1.427E-03$
	Std	$5.340E+00$	$6.941E+03$	$1.016E+00$	$1.423E+00$	$1.118E+00$	**5.339E−07**	$3.757E-06$	$1.034E-06$	$2.359E-05$	$2.389E-03$
	Rank	9	10	8	7	6	1	3	2	4	5
	Best	$1.198E+03$	$4.974E+01$	$2.106E-03$	$2.143E-04$	$1.476E-04$	$2.729E-05$	$9.448E-06$	$1.484E-05$	$9.049E-06$	**2.371E−06**
	Worst	$2.001E+03$	$4.241E+02$	$1.629E-02$	$1.567E-02$	$2.330E-02$	$1.299E-03$	$1.096E-03$	$3.918E-04$	$1.621E-03$	**3.207E−04**
F7	Ave	$1.518E+03$	$1.609E+02$	$7.634E-03$	$4.709E-03$	$5.183E-03$	$3.217E-04$	$3.569E-04$	$1.502E-04$	$4.565E-04$	**1.394E−04**
	Std	$1.889E+02$	$9.512E+01$	$3.085E-03$	$5.310E-03$	$5.977E-03$	$2.954E-04$	$2.842E-04$	$1.087E-04$	$4.189E-04$	**9.063E−05**
	Rank	10	9	8	6	7	3	4	2	5	1
	Best	$-1.781E+04$	$-7.934E+03$	$-1.999E+04$	$-4.187E+04$	$-4.190E+04$	**−4.190E+04**	$-2.698E+04$	**−4.190E+04**	**−4.190E+04**	**−4.190E+04**
	Worst	$-5.016E+03$	$-6.076E+03$	$-5.790E+03$	$-2.378E+04$	$-2.746E+04$	$-2.346E+04$	$-1.761E+04$	$-2.955E+04$	$-2.184E+04$	**−3.456E+04**
F8	Ave	$-1.050E+04$	$-6.759E+03$	$-1.623E+04$	$-3.407E+04$	$-3.358E+04$	$-3.150E+04$	$-2.276E+04$	$-3.604E+04$	$-3.379E+04$	**−4.050E+04**
	Std	$3.860E+03$	**4.827E+02**	$2.907E+03$	$5.642E+03$	$5.637E+03$	$6.075E+03$	$2.130E+03$	$4.310E+03$	$7.876E+03$	$2.173E+03$
	Rank	9	10	8	3	5	6	7	2	4	1

Table 6: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
	Best	$4.779E+02$	$5.265E+01$	$2.724E-09$	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Worst	$7.998E+02$	$4.466E+02$	$4.533E+01$	**0**	$1.137E-13$	**0**	**0**	**0**	**0**	**0**
F9	Ave	$6.303E+02$	$2.595E+02$	$3.979E+00$	**0**	$3.790E-15$	**0**	**0**	**0**	**0**	**0**
	Std	$8.286E+01$	$1.086E+02$	$8.599E+00$	**0**	$2.076E-14$	**0**	**0**	**0**	**0**	**0**
	Rank	10	9	8	1	7	1	1	1	1	1
	Best	$3.132E+00$	$9.221E+00$	$3.430E-07$	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**
	Worst	$4.305E+00$	$2.067E+01$	$1.375E-06$	$7.994E-15$	$7.994E-15$	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**	**8.882E – 16**
F10	Ave	$3.726E+00$	$1.899E+01$	$6.452E-07$	$4.322E-15$	$4.204E-15$	**0**	**0**	**0**	**0**	**0**
	Std	$2.946E-01$	$3.752E+00$	$2.326E-07$	$2.873E-15$	$2.273E-15$	**4.012E – 31**	**4.012E – 31**	**4.012E – 31**	**4.012E – 31**	**4.012E – 31**
	Rank	9	10	8	7	6	1	1	1	1	1
	Best	$2.761E-01$	$9.224E+00$	$4.273E-12$	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Worst	$6.430E-01$	$1.889E+02$	$2.807E-02$	$3.137E-01$	**0**	**0**	**0**	**0**	**0**	**0**
F11	Ave	$4.252E-01$	$8.541E+01$	$2.511E-03$	$1.046E-02$	**0**	**0**	**0**	**0**	**0**	**0**
	Std	$9.838E-02$	$5.334E+01$	$7.694E-03$	$5.728E-02$	**0**	**0**	**0**	**0**	**0**	**0**
	Rank	9	10	7	8	1	1	1	1	1	1
	Best	$9.845E-01$	$7.795E+07$	$1.679E-01$	$1.867E-02$	$1.876E-02$	$2.330E-20$	$1.123E-12$	$2.531E-12$	$3.574E-13$	**4.712E – 33**
	Worst	$9.589E+00$	$6.179E+08$	$3.925E-01$	$9.377E-02$	$1.440E-01$	**8.878E – 09**	$6.918E-08$	$1.562E-07$	$1.101E-08$	$9.480E-05$
F12	Ave	$4.297E+00$	$2.889E+08$	$2.624E-01$	$4.903E-02$	$4.792E-02$	**9.451E – 10**	$1.021E-08$	$1.261E-08$	$2.028E-09$	$2.722E-05$
	Std	$1.884E+00$	$1.390E+08$	$6.522E-02$	$1.946E-02$	$2.491E-02$	**1.909E – 09**	$1.627E-08$	$2.920E-08$	$2.960E-09$	$3.211E-05$
	Rank	9	10	8	7	6	1	3	4	2	5
	Best	$2.737E+01$	$2.004E+08$	$5.195E+00$	$1.224E+00$	$1.242E+00$	$6.058E-14$	$7.614E-10$	$6.218E-10$	$1.775E-10$	**1.350E – 32**
	Worst	$9.778E+01$	$1.593E+09$	$6.920E+00$	$5.151E+00$	$5.166E+00$	$4.220E-07$	$3.989E-06$	$1.629E-06$	$6.166E-06$	**1.350E – 32**
F13	Ave	$5.972E+01$	$6.758E+08$	$6.397E+00$	$2.787E+00$	$3.138E+00$	$4.654E-08$	$4.157E-07$	$2.250E-07$	$5.104E-07$	**1.350E – 32**
	Std	$1.998E+01$	$3.405E+08$	$4.285E-01$	$9.742E-01$	$8.932E-01$	$9.147E-08$	$8.210E-07$	$4.018E-07$	$1.269E-06$	**0**
	Rank	9	10	8	6	7	2	4	3	5	1
Average rank		8.92	9.62	7.46	6.38	6.15	3.31	3.46	1.70	2.85	1.61

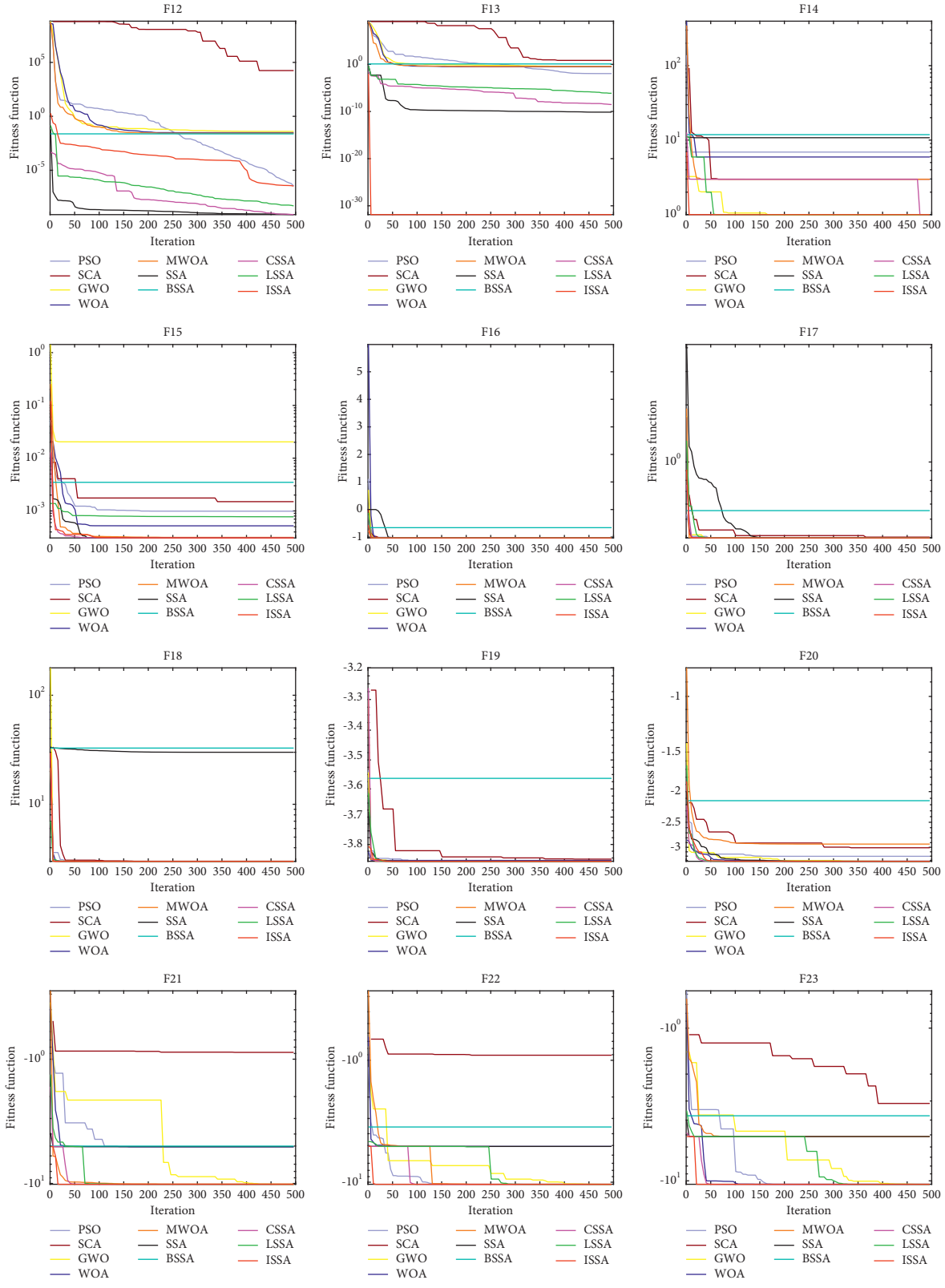
Table 7: Wilcoxon rank sum test results of each algorithm (30 dimensions).

F	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA
F1	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	NaN	$1.212E-12$	NaN	NaN
F2	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$4.057E-03$	$1.212E-12$	NaN	$1.212E-12$
F3	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$5.772E-11$	$1.212E-12$	NaN	$4.714E-05$
F4	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$4.574E-12$	$1.212E-12$
F5	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.211E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$
F6	$1.167E-02$	$2.954E-11$	$2.814E-10$	$2.954E-11$	$2.954E-11$	$1.957E-04$	$3.962E-04$	$3.540E-04$	$3.962E-04$
F7	$3.020E-11$	$3.020E-11$	$4.504E-11$	$9.063E-08$	$4.616E-10$	$1.023E-01$	$9.626E-02$	$7.062E-01$	$7.483E-02$
F8	$2.392E-11$	$2.392E-11$	$2.392E-11$	$2.831E-07$	$3.102E-08$	$1.783E-06$	$2.392E-11$	$3.327E-09$	$4.347E-08$
F9	$1.212E-12$	$1.212E-12$	$3.818E-12$	NaN	$3.337E-01$	NaN	NaN	NaN	NaN
F10	$1.211E-12$	$1.212E-12$	$1.199E-12$	$9.318E-08$	$1.317E-09$	NaN	NaN	NaN	NaN
F11	$1.212E-12$	$1.212E-12$	$6.617E-04$	$3.337E-01$	NaN	NaN	NaN	NaN	NaN
F12	$1.259E-01$	$2.982E-11$	$2.982E-11$	$2.982E-11$	$2.982E-11$	$9.460E-06$	$9.460E-06$	$9.460E-06$	$2.420E-05$
F13	$1.211E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$	$1.212E-12$
F14	$4.126E-07$	$3.800E-09$	$1.642E-05$	$2.328E-04$	$3.743E-05$	$5.429E-08$	$1.058E-10$	$1.608E-01$	$4.181E-01$
F15	$5.992E-11$	$2.982E-11$	$8.620E-09$	$1.081E-10$	$4.027E-11$	$9.080E-03$	$1.598E-01$	$3.241E-06$	$3.455E-03$
F16	$1.685E-14$	$3.620E-13$	$1.685E-14$	$1.685E-14$	$1.685E-14$	$2.708E-14$	$1.685E-14$	$1.685E-14$	$1.685E-14$
F17	NaN	$4.566E-12$	$3.337E-01$	$6.519E-04$	$6.500E-05$	NaN	NaN	NaN	NaN
F18	NaN	NaN	NaN	NaN	NaN	$1.607E-01$	NaN	NaN	NaN
F19	NaN	$1.129E-12$	$2.158E-02$	$1.828E-09$	$5.600E-10$	NaN	NaN	NaN	NaN
F20	$6.138E-01$	$6.738E-11$	$2.457E-02$	$1.021E-01$	$4.245E-04$	$1.000E+00$	$1.189E-01$	$5.271E-05$	$3.055E-01$
F21	$1.173E-05$	$1.212E-12$	$1.101E-02$	$1.878E-09$	$5.404E-11$	NaN	NaN	NaN	NaN
F22	$4.721E-03$	$1.720E-12$	$1.000E+00$	$1.704E-12$	$1.715E-12$	$2.416E-04$	$1.039E-10$	$3.337E-01$	$3.337E-01$
F23	$2.157E-02$	$1.212E-12$	$1.608E-01$	$1.208E-12$	$1.208E-12$	$2.773E-03$	$5.808E-09$	NaN	NaN



(a)

Figure 8: Continued.



(b)

Figure 8: Convergence diagram of each algorithm.

smallest, followed by LSSA, BSSA, and SSA. As shown in Figure 8, the ISSA shows excellent optimization speed and convergence accuracy, and the convergence speed is fastest in most functions. The ISSA converges faster in the single-peak benchmark function and has better ability to resist the attraction local optimum in the multipeak benchmark function.

5. CEC 2017 Function Test

In order to better illustrate the generality and effectiveness of the algorithm and avoid that the ISSA is only applicable to the case where the optimal value is 0, the algorithm is tested on the CEC 2017 test function. The evaluation times are $10000 * \text{dim}$, the number of population individuals is 30, the dimension is 30, SD is set to 0.6, and other parameters remain unchanged. In this paper, the above algorithms are run independently for 30 times, and five indexes of each algorithm are calculated according to the results, namely, the best value (Best), the worst value (Worst), the median (Med), the average value (Ave), and the standard deviation (Std). Finally, each algorithm will be ranked according to the average value of the algorithm in the function. The optimal value of each index is treated in bold. Due to the defects of F2 function, it will not be tested in this paper. The specific test results are shown in Table 8. At the same time, six functions F4, F7, F14, F17, F24, and F27 are selected to draw the box diagram of the results, as shown in Figure 9.

It can be seen from the data in Table 8 that, in the 29 functions, ISSA ranks first in most functions and its average ranking is better than other algorithms. ISSA has good optimization effect and can be close to the theoretical optimal value of each function. In F1, F3, F6, F7, F8, F13,F14, F18, and F29, each index of ISSA is the best in the algorithm. Among the 20 functions, ISSA finds the optimal value of all algorithms, which shows that ISSA has strong optimization ability. When the optimal solution is not 0, ISSA shows better optimization performance than SSA, and PSO and CSSA also show good performance.

As can be seen from the box diagram in Figure 9, the ISSA has strong search ability and is closest to the theoretical optimal value among all algorithms. And ISSA's box graph has shorter length and stronger stability than other algorithms. Among them, the realization of SSA is poor because the individual in the SSA is directly jumping when approaching the current optimal solution, rather than moving to the current optimal solution like PSO. This problem leads to the rapid convergence of SSA, but it is easy to miss the high-quality solution and fall into local optimization. The ISSA uses the improved strategy to make up for this disadvantage, makes full use of the current solution, ensures the convergence speed, and increases its ability to jump out of the local optimum. But overall, the ISSA has better optimization performance than other algorithms, has good universality and effectiveness, and can adapt to some complex optimization problems.

6. PID Parameter Tuning

A PID controller is the most widely used controller in the industry (accounting for about 90% of the controller). The PID controller is composed of three basic gain parameters to control the controlled object. It is mainly applicable to the system whose basic linear and dynamic characteristics do not change with time. Its structure is shown in Figure 10. When the proportional parameter K_p increases, the rise time and steady-state error decrease. When the integral parameter K_i increases, the rise time is smaller, but the stability time and overshoot increase. The negative effect of the K_i increase can be overcome by adjusting the differential parameter K_d. The relationship between output and input of the PID controller is as follows:

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}. \tag{28}$$

Manual PID parameter tuning is a time-consuming process. Generally, it is tried through the experience and skills of engineers and the intelligent algorithm can complete the parameter tuning in a short time. In order to verify the practicability of ISSA, this paper uses ISSA to optimize PID parameters, simulates under unit step response and sinusoidal input response respectively, and tests with SSA to prove its optimization performance.

In this paper, the objective function [39] is set as follows:

$$F = \int_0^{\infty} \left(w_1 |e(t)| + w_2 u^2(t)\right) \mathrm{d}t, \tag{29}$$

$e(t)$ is the error between the input value and the output value. Considering the dynamic characteristics of the iterative process, the integral of its absolute value is adopted; $u(t)$ is the control value, which is added to avoid excessive control range; w_1 and w_2 are weights, and the value range is [0, 1]. In addition, measures shall be taken to prevent overshoot, that is, when overshoot occurs, an additional overshoot item shall be introduced into the objective function. At this time, the settings are as follows:

$$F = \int_0^{\infty} \left(w_1 |e(t)| + w_2 u^2(t) + w_3 |e(t)|\right) \mathrm{d}t, \quad e(t) < 0, \tag{30}$$

where w_3 is the weight and $w_3 \gg w_1$. Generally, $w_1 = 0.999, w_2 = 0.001, w_3 = 100$. Therefore, the goal of ISSA is to find a set of PID parameters to minimize the error of objective function. In this paper, the number of individuals and the number of iterations of the population are set to be 30 and 50, respectively. Other parameters are consistent with those of Table 2. The test is carried out under the condition of unit leap forward and sinusoidal input respectively and run independently for 10 times. The test results are shown in Figure 11, and the optimization results are shown in Table 9.

Since the results of 10 independent runs are consistent, only one is shown here. It can be seen from Figure 11 and Table 9 that in the unit step response, ISSA can complete the parameter setting in a very short time, and the convergence

Table 8: Test results of each algorithm in CEC 2017.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
F1	Best	1.52E + 02	9.06E + 09	5.10E + 08	4.33E + 06	4.28E + 09	1.09E + 02	1.65E + 07	1.23E + 02	1.06E + 02	1.00E + 02
	Worst	9.74E + 03	1.84E + 10	3.22E + 09	1.17E + 08	1.04E + 10	1.99E + 04	1.66E + 08	1.20E + 04	2.08E + 04	**9.24E + 03**
	Med	1.68E + 03	1.19E + 10	1.05E + 09	2.06E + 07	6.36E + 09	6.79E + 02	8.20E + 07	1.48E + 03	5.36E + 03	**5.23E + 02**
	Ave	2.53E + 03	1.20E + 10	1.35E + 09	2.99E + 07	6.43E + 09	3.47E + 03	9.87E + 07	3.18E + 03	8.01E + 03	**1.78E + 03**
	Std	2.61E + 03	2.06E + 09	7.53E + 08	2.74E + 07	1.21E + 09	5.82E + 03	4.14E + 07	3.29E + 03	7.96E + 03	**2.32E + 03**
	Rank	2	10	8	6	9	4	7	3	5	1
F3	Best	1.95E + 03	2.30E + 04	2.87E + 04	8.52E + 04	4.82E + 04	1.26E + 04	4.67E + 04	5.46E + 03	2.15E + 04	**3.07E + 02**
	Worst	1.19E + 04	4.89E + 04	6.27E + 04	3.02E + 05	6.49E + 04	2.98E + 04	6.87E + 04	1.44E + 04	4.99E + 04	**7.66E + 02**
	Med	5.12E + 03	3.46E + 04	5.06E + 04	1.89E + 05	5.56E + 04	1.67E + 04	5.67E + 04	1.02E + 04	3.30E + 04	**4.90E + 02**
	Ave	5.49E + 03	3.39E + 04	4.92E + 04	2.00E + 05	5.53E + 04	1.78E + 04	5.69E + 04	9.94E + 03	3.21E + 04	**4.64E + 02**
	Std	1.58E + 03	8.48E + 03	1.15E + 04	7.55E + 04	3.44E + 03	4.49E + 03	8.03E + 03	2.35E + 03	7.20E + 03	**1.20E + 02**
	Rank	2	6	7	10	8	4	9	3	5	1
F4	Best	4.18E + 02	8.86E + 02	5.20E + 02	5.03E + 02	1.23E + 03	4.59E + 02	5.10E + 02	4.59E + 02	4.08E + 02	**4.04E + 02**
	Worst	**4.82E + 02**	1.89E + 03	6.49E + 02	6.48E + 02	2.73E + 03	5.38E + 02	6.93E + 02	5.35E + 02	5.40E + 02	6.09E + 02
	Med	4.79E + 02	1.53E + 03	6.18E + 02	5.58E + 02	2.13E + 03	5.10E + 02	5.80E + 02	4.88E + 02	**4.72E + 02**	4.77E + 02
	Ave	**4.63E + 02**	1.48E + 03	6.09E + 02	5.61E + 02	2.08E + 03	5.08E + 02	5.78E + 02	4.83E + 02	4.82E + 02	4.66E + 02
	Std	2.67E + 01	2.49E + 02	3.67E + 01	4.26E + 01	4.82E + 02	1.79E + 01	5.62E + 01	**1.56E + 01**	2.91E + 01	4.15E + 01
	Rank	1	9	8	6	10	5	7	4	3	2
F5	Best	6.31E + 02	7.39E + 02	**5.53E + 02**	7.05E + 02	7.99E + 02	7.34E + 02	5.91E + 02	6.56E + 02	6.21E + 02	6.49E + 02
	Worst	7.30E + 02	8.27E + 02	7.47E + 02	9.26E + 02	9.37E + 02	8.20E + 02	**6.70E + 02**	8.07E + 02	8.21E + 02	8.16E + 02
	Med	7.00E + 02	7.62E + 02	**5.94E + 02**	7.57E + 02	8.49E + 02	8.15E + 02	6.45E + 02	7.40E + 02	7.18E + 02	7.80E + 02
	Ave	6.90E + 02	7.68E + 02	**6.03E + 02**	7.80E + 02	8.69E + 02	8.08E + 02	6.34E + 02	7.40E + 02	7.32E + 02	7.62E + 02
	Std	3.06E + 01	**1.73E + 01**	3.50E + 01	7.36E + 01	4.91E + 01	1.79E + 01	2.40E + 01	4.57E + 01	5.45E + 01	3.41E + 01
	Rank	3	7	1	8	10	9	2	5	4	6
F6	Best	6.34E + 02	6.42E + 02	6.38E + 02	6.61E + 02	6.65E + 02	6.57E + 02	6.08E + 02	6.21E + 02	6.22E + 02	**6.02E + 02**
	Worst	6.57E + 02	6.61E + 02	6.68E + 02	6.83E + 02	6.88E + 02	6.78E + 02	6.65E + 02	6.57E + 02	6.58E + 02	**6.12E + 02**
	Med	6.45E + 02	6.50E + 02	6.57E + 02	6.70E + 02	6.76E + 02	6.64E + 02	6.29E + 02	6.27E + 02	6.41E + 02	**6.11E + 02**
	Ave	6.46E + 02	6.49E + 02	6.56E + 02	6.72E + 02	6.74E + 02	6.64E + 02	6.38E + 02	6.33E + 02	6.41E + 02	**6.08E + 02**
	Std	7.80E + 00	5.44E + 00	6.90E + 00	8.86E + 00	5.46E + 00	6.39E + 00	2.16E + 01	1.08E + 01	1.02E + 01	**4.37E + 00**
	Rank	5	6	7	9	10	8	3	2	4	1
F7	Best	1.07E + 03	1.08E + 03	8.20E + 02	1.06E + 03	1.19E + 03	1.18E + 03	8.74E + 02	9.28E + 02	8.64E + 02	**7.99E + 02**
	Worst	1.37E + 03	1.18E + 03	9.91E + 02	1.44E + 03	1.41E + 03	1.35E + 03	1.33E + 03	1.34E + 03	1.33E + 03	**8.76E + 02**
	Med	1.32E + 03	1.10E + 03	8.91E + 02	1.24E + 03	1.27E + 03	1.33E + 03	9.82E + 02	1.17E + 03	1.14E + 03	**8.51E + 02**
	Ave	1.29E + 03	1.11E + 03	8.89E + 02	1.28E + 03	1.29E + 03	1.32E + 03	1.00E + 03	1.17E + 03	1.14E + 03	**8.46E + 02**
	Std	6.72E + 01	2.99E + 01	4.61E + 01	1.15E + 02	7.20E + 01	3.85E + 01	8.42E + 01	1.09E + 02	1.44E + 02	**1.66E + 01**
	Rank	8	4	2	7	9	10	3	6	5	1
F8	Best	8.74E + 02	1.02E + 03	9.08E + 02	9.42E + 02	1.04E + 03	9.38E + 02	9.15E + 02	8.60E + 02	8.84E + 02	**8.55E + 02**
	Worst	9.62E + 02	1.09E + 03	1.01E + 03	1.08E + 03	1.11E + 03	1.05E + 03	9.79E + 02	9.82E + 02	1.01E + 03	**9.14E + 02**
	Med	9.15E + 02	1.04E + 03	9.52E + 02	9.72E + 02	1.09E + 03	9.92E + 02	9.59E + 02	9.43E + 02	9.66E + 02	**8.80E + 02**
	Ave	9.21E + 02	1.05E + 03	9.57E + 02	9.89E + 02	1.08E + 03	9.94E + 02	9.53E + 02	9.48E + 02	9.58E + 02	**8.82E + 02**
	Std	2.58E + 01	2.28E + 01	2.31E + 01	3.69E + 01	1.63E + 01	2.56E + 01	1.44E + 01	2.79E + 01	3.98E + 01	**1.37E + 01**
	Rank	2	9	5	7	10	8	4	3	6	1

Table 8: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
F9	Best	2.45E + 03	3.74E + 03	3.57E + 03	3.60E + 03	6.34E + 03	4.01E + 03	2.04E + 03	3.08E + 03	5.06E + 03	**1.06E + 03**
	Worst	5.01E + 03	7.85E + 03	5.79E + 03	1.35E + 04	1.15E + 04	5.58E + 03	9.94E + 03	5.44E + 03	5.56E + 03	**1.85E + 03**
	Med	3.99E + 03	4.43E + 03	4.97E + 03	6.12E + 03	8.81E + 03	5.41E + 03	5.85E + 03	5.35E + 03	5.39E + 03	**1.44E + 03**
	Ave	3.84E + 03	4.95E + 03	4.95E + 03	6.82E + 03	8.73E + 03	5.30E + 03	6.02E + 03	4.88E + 03	5.37E + 03	**1.42E + 03**
	Std	5.34E + 02	1.09E + 03	4.57E + 02	1.82E + 03	1.34E + 03	3.75E + 02	2.32E + 03	8.19E + 02	**1.23E + 02**	1.87E + 02
	Rank	2	4	5	9	10	6	8	3	7	1
F10	Best	3.71E + 03	7.33E + 03	**3.15E + 03**	5.34E + 03	5.74E + 03	4.57E + 03	4.82E + 03	4.09E + 03	3.91E + 03	4.06E + 03
	Worst	5.45E + 03	8.48E + 03	**5.06E + 03**	8.13E + 03	8.59E + 03	9.10E + 03	9.80E + 03	6.33E + 03	6.53E + 03	8.89E + 03
	Med	4.27E + 03	8.17E + 03	**4.20E + 03**	7.29E + 03	7.34E + 03	5.95E + 03	6.57E + 03	5.39E + 03	4.58E + 03	6.24E + 03
	Ave	4.47E + 03	8.13E + 03	**4.05E + 03**	7.00E + 03	7.54E + 03	6.43E + 03	6.71E + 03	5.42E + 03	4.83E + 03	6.09E + 03
	Std	5.13E + 02	**2.79E + 02**	6.38E + 02	9.77E + 02	6.40E + 02	1.63E + 03	1.60E + 03	5.13E + 02	6.99E + 02	1.09E + 03
	Rank	2	10	1	8	9	6	7	4	3	5
F11	Best	1.16E + 03	1.71E + 03	1.34E + 03	1.47E + 03	2.57E + 03	1.19E + 03	1.31E + 03	1.16E + 03	1.25E + 03	**1.14E + 03**
	Worst	**1.25E + 03**	3.73E + 03	2.12E + 03	2.49E + 03	5.28E + 03	1.39E + 03	1.88E + 03	1.32E + 03	1.59E + 03	1.29E + 03
	Med	1.19E + 03	1.90E + 03	1.44E + 03	2.30E + 03	4.37E + 03	1.23E + 03	1.43E + 03	1.26E + 03	1.44E + 03	**1.19E + 03**
	Ave	**1.19E + 03**	2.02E + 03	1.59E + 03	2.12E + 03	4.23E + 03	1.25E + 03	1.45E + 03	1.25E + 03	1.43E + 03	1.19E + 03
	Std	**2.46E + 01**	4.01E + 02	2.58E + 02	3.84E + 02	6.64E + 02	4.68E + 01	1.36E + 02	4.06E + 01	9.29E + 01	3.12E + 01
	Rank	1	8	7	9	10	3	6	4	5	2
F12	Best	**2.20E + 04**	6.42E + 08	4.00E + 06	1.03E + 07	6.90E + 08	3.78E + 04	9.49E + 05	1.09E + 05	1.25E + 05	2.90E + 04
	Worst	1.25E + 06	1.99E + 09	3.22E + 08	1.08E + 08	2.59E + 09	1.13E + 07	1.22E + 07	**6.57E + 05**	9.09E + 05	1.91E + 07
	Med	3.54E + 05	8.82E + 08	3.93E + 07	6.62E + 07	9.19E + 08	2.07E + 06	2.91E + 06	**2.08E + 05**	4.45E + 05	1.54E + 06
	Ave	4.22E + 05	1.03E + 09	6.63E + 07	6.01E + 07	1.11E + 09	2.45E + 06	4.41E + 06	**2.23E + 05**	4.57E + 05	2.11E + 06
	Std	3.55E + 05	4.03E + 08	8.44E + 07	2.84E + 07	5.40E + 08	2.66E + 06	3.09E + 06	**1.11E + 05**	2.80E + 05	3.55E + 06
	Rank	2	9	8	7	10	5	6	1	3	4
F13	Best	2.90E + 03	2.13E + 08	3.83E + 04	3.85E + 04	6.22E + 07	3.05E + 03	8.09E + 03	2.97E + 03	5.95E + 03	**1.68E + 03**
	Worst	5.62E + 04	6.13E + 08	1.40E + 08	2.82E + 05	4.32E + 08	7.25E + 04	4.51E + 06	7.12E + 04	5.45E + 04	**1.96E + 04**
	Med	**8.39E + 03**	3.48E + 08	1.84E + 05	1.22E + 05	1.53E + 08	1.39E + 04	2.34E + 04	9.69E + 03	2.75E + 04	9.36E + 03
	Ave	1.21E + 04	3.69E + 08	3.90E + 07	1.19E + 05	1.63E + 08	2.56E + 04	5.41E + 05	1.38E + 04	2.92E + 04	**9.11E + 03**
	Std	1.07E + 04	9.87E + 07	5.71E + 07	5.05E + 04	8.14E + 07	2.50E + 04	1.36E + 06	1.58E + 04	2.09E + 04	**4.37E + 03**
	Rank	2	10	8	6	9	4	7	3	5	1
F14	Best	4.96E + 03	4.08E + 04	2.93E + 03	1.37E + 04	1.41E + 06	5.09E + 03	7.46E + 03	1.67E + 03	2.50E + 03	**1.66E + 03**
	Worst	1.12E + 05	2.71E + 05	8.84E + 05	7.05E + 06	4.40E + 06	1.64E + 05	1.79E + 06	7.66E + 04	1.09E + 05	**5.18E + 04**
	Med	2.71E + 04	8.97E + 04	4.63E + 04	1.11E + 06	2.68E + 06	3.07E + 04	3.35E + 05	1.39E + 04	2.86E + 04	**5.97E + 03**
	Ave	3.93E + 04	1.25E + 05	2.31E + 05	1.86E + 06	2.54E + 06	3.37E + 04	4.24E + 05	1.69E + 04	3.86E + 04	**1.52E + 04**
	Std	2.72E + 04	6.22E + 04	2.90E + 05	1.72E + 06	9.81E + 05	2.81E + 04	4.18E + 05	1.72E + 04	3.18E + 04	**1.71E + 04**
	Rank	5	6	7	9	10	3	8	2	4	1
F15	Best	1.65E + 03	6.16E + 05	1.39E + 04	2.15E + 04	3.22E + 06	2.08E + 03	2.30E + 03	1.85E + 03	1.86E + 03	**1.57E + 03**
	Worst	2.12E + 04	5.07E + 07	5.42E + 05	1.83E + 05	5.30E + 07	2.29E + 04	2.04E + 04	**1.50E + 04**	4.43E + 04	3.46E + 04
	Med	4.05E + 03	5.21E + 06	1.03E + 05	4.40E + 04	7.88E + 06	5.81E + 03	6.81E + 03	3.46E + 03	2.92E + 04	**2.38E + 03**
	Ave	5.39E + 03	1.11E + 07	1.12E + 05	6.36E + 04	1.36E + 07	7.88E + 03	6.92E + 03	**4.75E + 03**	2.35E + 04	4.92E + 03
	Std	5.43E + 03	1.08E + 07	1.23E + 05	4.40E + 04	1.44E + 07	6.35E + 03	3.96E + 03	**3.14E + 03**	1.80E + 04	6.67E + 03
	Rank	3	9	8	7	10	5	4	1	6	2

Table 8: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
F16	Best	2.24E + 03	3.37E + 03	1.98E + 03	2.93E + 03	3.86E + 03	2.83E + 03	2.19E + 03	2.42E + 03	2.33E + 03	**1.88E + 03**
	Worst	**3.06E + 03**	4.09E + 03	3.12E + 03	4.56E + 03	5.69E + 03	8.55E + 03	3.60E + 03	3.30E + 03	3.33E + 03	5.57E + 03
	Med	2.85E + 03	3.66E + 03	**2.77E + 03**	3.33E + 03	4.15E + 03	3.79E + 03	2.81E + 03	3.10E + 03	2.89E + 03	3.23E + 03
	Ave	2.80E + 03	3.68E + 03	**2.56E + 03**	3.44E + 03	4.31E + 03	4.12E + 03	2.84E + 03	3.00E + 03	2.87E + 03	3.35E + 03
	Std	2.70E + 02	**1.55E + 02**	4.02E + 02	4.03E + 02	4.33E + 02	1.23E + 03	3.20E + 02	2.71E + 02	2.51E + 02	1.09E + 03
	Rank	2	8	1	7	10	9	3	5	4	6
F17	Best	1.94E + 03	2.11E + 03	2.07E + 03	2.15E + 03	2.22E + 03	1.99E + 03	1.96E + 03	2.05E + 03	2.13E + 03	**1.83E + 03**
	Worst	2.75E + 03	2.69E + 03	3.81E + 03	3.22E + 03	3.15E + 03	3.36E + 03	2.92E + 03	3.01E + 03	3.02E + 03	**2.25E + 03**
	Med	2.32E + 03	2.30E + 03	3.08E + 03	2.56E + 03	2.95E + 03	2.68E + 03	2.40E + 03	2.44E + 03	2.65E + 03	**1.99E + 03**
	Ave	2.33E + 03	2.37E + 03	3.00E + 03	2.65E + 03	2.89E + 03	2.81E + 03	2.42E + 03	2.46E + 03	2.58E + 03	**2.01E + 03**
	Std	2.55E + 02	**1.44E + 02**	4.67E + 02	2.65E + 02	2.49E + 02	3.40E + 02	2.45E + 02	2.11E + 02	2.25E + 02	1.48E + 02
	Rank	2	3	10	7	9	8	4	5	6	1
F18	Best	5.23E + 04	3.20E + 05	4.38E + 04	2.75E + 05	2.86E + 06	1.19E + 04	5.88E + 04	3.78E + 04	1.09E + 05	**6.74E + 03**
	Worst	5.93E + 05	6.66E + 06	1.06E + 07	8.14E + 06	2.41E + 07	1.54E + 06	7.28E + 06	4.14E + 05	1.47E + 06	**1.74E + 05**
	Med	3.46E + 05	1.48E + 06	4.59E + 05	2.96E + 06	8.73E + 06	7.18E + 04	5.67E + 05	8.42E + 04	2.53E + 05	**4.00E + 04**
	Ave	2.74E + 05	1.83E + 06	1.55E + 06	2.81E + 06	8.77E + 06	1.80E + 05	1.79E + 06	1.43E + 05	3.56E + 05	**7.38E + 04**
	Std	1.84E + 05	1.65E + 06	2.78E + 06	2.01E + 06	3.77E + 06	3.80E + 05	2.81E + 06	1.22E + 05	3.37E + 05	**6.25E + 04**
	Rank	4	8	6	9	10	3	7	2	5	1
F19	Best	2.04E + 03	4.15E + 06	2.06E + 04	9.08E + 05	6.49E + 06	2.53E + 03	2.57E + 03	2.17E + 03	2.36E + 03	**2.02E + 03**
	Worst	2.18E + 04	3.99E + 07	4.22E + 06	1.30E + 07	6.12E + 07	1.63E + 04	2.10E + 04	**9.56E + 03**	5.64E + 04	2.64E + 04
	Med	**3.82E + 03**	2.04E + 07	5.12E + 05	3.18E + 06	1.39E + 07	5.51E + 03	4.44E + 03	5.15E + 03	6.78E + 03	6.49E + 03
	Ave	**5.54E + 03**	2.38E + 07	9.64E + 05	4.52E + 06	2.01E + 07	6.91E + 03	7.75E + 03	5.63E + 03	1.09E + 04	7.79E + 03
	Std	4.49E + 03	9.96E + 06	1.34E + 06	4.12E + 06	1.30E + 07	3.76E + 03	6.79E + 03	**2.43E + 03**	1.36E + 04	5.90E + 03
	Rank	1	10	7	8	9	3	4	2	6	5
F20	Best	2.26E + 03	2.38E + 03	2.27E + 03	2.38E + 03	2.49E + 03	2.39E + 03	**2.14E + 03**	2.29E + 03	2.30E + 03	2.15E + 03
	Worst	2.89E + 03	**2.73E + 03**	3.39E + 03	3.15E + 03	2.99E + 03	3.43E + 03	3.22E + 03	2.95E + 03	2.91E + 03	2.77E + 03
	Med	2.44E + 03	2.58E + 03	3.04E + 03	2.80E + 03	2.78E + 03	2.75E + 03	2.66E + 03	2.54E + 03	2.71E + 03	**2.27E + 03**
	Ave	2.49E + 03	2.59E + 03	2.97E + 03	2.74E + 03	2.78E + 03	2.84E + 03	2.60E + 03	2.52E + 03	2.69E + 03	**2.30E + 03**
	Std	1.63E + 02	**8.77E + 01**	2.43E + 02	2.01E + 02	1.16E + 02	2.39E + 02	1.92E + 02	1.83E + 02	1.27E + 02	1.30E + 02
	Rank	2	4	10	7	8	9	5	3	6	1
F21	Best	2.39E + 03	2.52E + 03	2.42E + 03	2.50E + 03	2.56E + 03	2.49E + 03	2.40E + 03	**2.20E + 03**	2.42E + 03	2.36E + 03
	Worst	2.51E + 03	2.60E + 03	2.68E + 03	2.68E + 03	2.74E + 03	2.73E + 03	2.49E + 03	2.58E + 03	2.58E + 03	**2.46E + 03**
	Med	2.45E + 03	2.53E + 03	2.54E + 03	2.57E + 03	2.63E + 03	2.62E + 03	2.43E + 03	2.49E + 03	2.46E + 03	**2.39E + 03**
	Ave	2.45E + 03	2.54E + 03	2.53E + 03	2.58E + 03	2.63E + 03	2.59E + 03	2.44E + 03	2.48E + 03	2.48E + 03	**2.41E + 03**
	Std	2.32E + 01	**1.77E + 01**	5.36E + 01	5.01E + 01	3.40E + 01	5.86E + 01	3.25E + 01	6.21E + 01	4.72E + 01	3.81E + 01
	Rank	3	7	6	8	10	9	2	4	5	1
F22	Best	**2.30E + 03**	3.96E + 03	2.50E + 03	2.33E + 03	3.59E + 03	2.38E + 03	2.34E + 03	**2.30E + 03**	**2.30E + 03**	2.38E + 03
	Worst	**6.81E + 03**	9.97E + 03	1.00E + 04	8.80E + 03	1.03E + 04	1.04E + 04	8.79E + 03	7.87E + 03	7.58E + 03	9.42E + 03
	Med	5.86E + 03	9.60E + 03	4.99E + 03	7.17E + 03	4.34E + 03	7.44E + 03	2.38E + 03	**2.30E + 03**	5.73E + 03	7.31E + 03
	Ave	4.91E + 03	9.00E + 03	4.48E + 03	6.68E + 03	6.02E + 03	7.56E + 03	**3.30E + 03**	3.34E + 03	5.46E + 03	7.27E + 03
	Std	1.90E + 03	1.71E + 03	1.98E + 03	1.76E + 03	2.55E + 03	1.42E + 03	2.12E + 03	1.94E + 03	2.08E + 03	**1.30E + 03**
	Rank	4	10	3	7	6	9	1	2	5	8

Table 8: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
F23	Best	**2.70E + 03**	2.95E + 03	2.71E + 03	2.93E + 03	2.99E + 03	2.99E + 03	2.76E + 03	2.83E + 03	2.78E + 03	2.85E + 03
	Worst	3.36E + 03	3.05E + 03	2.95E + 03	3.29E + 03	3.25E + 03	3.57E + 03	**2.91E + 03**	3.06E + 03	2.95E + 03	3.59E + 03
	Med	3.15E + 03	2.97E + 03	**2.75E + 03**	3.03E + 03	3.05E + 03	3.33E + 03	2.83E + 03	2.92E + 03	2.82E + 03	3.31E + 03
	Ave	3.16E + 03	2.98E + 03	**2.77E + 03**	3.02E + 03	3.10E + 03	3.33E + 03	2.82E + 03	2.93E + 03	2.83E + 03	3.26E + 03
	Std	1.45E + 02	**3.02E + 01**	5.08E + 01	9.83E + 01	9.88E + 01	1.35E + 02	4.96E + 01	6.17E + 01	4.42E + 01	1.70E + 02
	Rank	8	5	1	6	7	10	2	4	3	9
F24	Best	3.07E + 03	3.13E + 03	3.11E + 03	3.00E + 03	3.13E + 03	3.19E + 03	2.91E + 03	2.98E + 03	2.93E + 03	**2.87E + 03**
	Worst	3.40E + 03	3.20E + 03	3.77E + 03	3.36E + 03	3.31E + 03	3.69E + 03	**3.06E + 03**	3.37E + 03	3.10E + 03	3.08E + 03
	Med	3.23E + 03	3.17E + 03	3.34E + 03	3.19E + 03	3.20E + 03	3.34E + 03	3.01E + 03	3.13E + 03	3.01E + 03	**2.91E + 03**
	Ave	3.24E + 03	3.17E + 03	3.40E + 03	3.17E + 03	3.21E + 03	3.33E + 03	2.99E + 03	3.14E + 03	3.01E + 03	**2.93E + 03**
	Std	1.33E + 02	**1.72E + 01**	1.94E + 02	7.64E + 01	5.73E + 01	1.05E + 02	4.18E + 01	1.03E + 02	4.47E + 01	5.27E + 01
	Rank	8	5	10	6	7	9	2	4	3	1
F25	Best	2.88E + 03	3.11E + 03	2.92E + 03	2.91E + 03	3.30E + 03	2.88E + 03	2.90E + 03	2.88E + 03	2.88E + 03	**2.88E + 03**
	Worst	2.90E + 03	3.36E + 03	3.03E + 03	3.05E + 03	3.55E + 03	2.95E + 03	3.02E + 03	2.94E + 03	2.89E + 03	2.95E + 03
	Med	**2.88E + 03**	3.22E + 03	2.98E + 03	2.93E + 03	3.39E + 03	2.90E + 03	2.96E + 03	2.89E + 03	2.89E + 03	2.89E + 03
	Ave	**2.88E + 03**	3.23E + 03	2.98E + 03	2.95E + 03	3.38E + 03	2.91E + 03	2.97E + 03	2.90E + 03	2.89E + 03	2.90E + 03
	Std	3.67E + 00	6.72E + 01	3.00E + 01	3.94E + 01	5.73E + 01	2.12E + 01	3.70E + 01	1.78E + 01	1.51E + 00	2.86E + 01
	Rank	1	9	8	6	10	5	7	3	2	4
F26	Best	5.76E + 03	6.49E + 03	4.29E + 03	6.03E + 03	7.32E + 03	3.01E + 03	3.55E + 03	2.90E + 03	2.80E + 03	**2.80E + 03**
	Worst	8.91E + 03	7.27E + 03	**5.10E + 03**	9.96E + 03	1.04E + 04	1.21E + 04	7.60E + 03	7.55E + 03	6.22E + 03	1.09E + 04
	Med	6.60E + 03	6.97E + 03	**4.63E + 03**	7.58E + 03	8.11E + 03	9.02E + 03	5.23E + 03	5.60E + 03	5.13E + 03	6.32E + 03
	Ave	7.08E + 03	6.92E + 03	**4.64E + 03**	7.58E + 03	8.29E + 03	8.76E + 03	5.23E + 03	5.65E + 03	4.67E + 03	6.69E + 03
	Std	1.06E + 03	1.75E + 02	**1.51E + 02**	6.53E + 02	6.15E + 02	1.80E + 03	7.03E + 02	1.07E + 03	1.35E + 03	2.08E + 03
	Rank	7	6	1	8	9	10	3	4	2	5
F27	Best	**3.17E + 03**	3.34E + 03	3.22E + 03	3.27E + 03	3.38E + 03	3.33E + 03	3.22E + 03	3.23E + 03	3.22E + 03	3.20E + 03
	Worst	3.42E + 03	3.49E + 03	3.33E + 03	3.68E + 03	4.00E + 03	4.14E + 03	3.40E + 03	3.41E + 03	3.35E + 03	**3.20E + 03**
	Med	**3.19E + 03**	3.41E + 03	3.25E + 03	3.33E + 03	3.60E + 03	3.80E + 03	3.28E + 03	3.30E + 03	3.23E + 03	3.20E + 03
	Ave	3.21E + 03	3.41E + 03	3.26E + 03	3.41E + 03	3.61E + 03	3.71E + 03	3.29E + 03	3.32E + 03	3.25E + 03	**3.20E + 03**
	Std	6.02E + 01	3.97E + 01	2.94E + 01	1.39E + 02	1.26E + 02	2.21E + 02	6.50E + 01	6.10E + 01	4.13E + 01	**1.09E − 04**
	Rank	2	7	4	8	9	10	5	6	3	1
F28	Best	3.10E + 03	3.65E + 03	3.30E + 03	3.27E + 03	3.66E + 03	3.20E + 03	3.26E + 03	3.10E + 03	3.12E + 03	**3.10E + 03**
	Worst	3.27E + 03	4.12E + 03	3.62E + 03	3.40E + 03	4.26E + 03	3.29E + 03	3.37E + 03	3.26E + 03	**3.26E + 03**	3.30E + 03
	Med	3.21E + 03	3.82E + 03	3.41E + 03	3.33E + 03	3.87E + 03	3.22E + 03	3.33E + 03	**3.11E + 03**	3.21E + 03	3.30E + 03
	Ave	3.21E + 03	3.85E + 03	3.40E + 03	3.33E + 03	3.95E + 03	3.22E + 03	3.33E + 03	**3.15E + 03**	3.22E + 03	3.28E + 03
	Std	4.84E + 01	1.55E + 02	7.88E + 01	2.67E + 01	1.72E + 02	**2.45E + 01**	3.22E + 01	6.51E + 01	3.33E + 01	4.67E + 01
	Rank	2	9	8	6	10	3	7	1	4	5
F29	Best	3.60E + 03	4.24E + 03	3.94E + 03	4.17E + 03	4.73E + 03	4.27E + 03	3.59E + 03	**3.42E + 03**	3.52E + 03	3.58E + 03
	Worst	4.30E + 03	5.05E + 03	8.78E + 03	5.73E + 03	6.02E + 03	7.31E + 03	4.26E + 03	4.82E + 03	4.40E + 03	**4.06E + 03**
	Med	3.95E + 03	4.53E + 03	4.92E + 03	4.97E + 03	5.11E + 03	5.24E + 03	4.16E + 03	4.09E + 03	4.05E + 03	**3.78E + 03**
	Ave	3.94E + 03	4.59E + 03	5.22E + 03	4.88E + 03	5.22E + 03	5.27E + 03	4.06E + 03	4.27E + 03	3.97E + 03	**3.79E + 03**
	Std	1.63E + 02	2.42E + 02	9.10E + 02	3.98E + 02	3.54E + 02	6.08E + 02	1.77E + 02	4.26E + 02	2.14E + 02	**1.27E + 02**
	Rank	2	6	8	7	9	10	4	5	3	1

Table 8: Continued.

F	Index	PSO	SCA	GWO	WOA	MWOA	SSA	BSSA	CSSA	LSSA	ISSA
F30	Best	5.40$E+03$	4.89$E+07$	1.44$E+06$	4.12$E+06$	2.32$E+07$	1.21$E+04$	1.11$E+04$	5.82$E+03$	6.80$E+03$	**3.61E+03**
	Worst	**1.07E+04**	1.15$E+08$	3.20$E+07$	4.34$E+07$	3.98$E+08$	6.06$E+06$	3.05$E+05$	1.88$E+04$	2.42$E+04$	1.29$E+05$
	Med	**6.25E+03**	7.19$E+07$	7.15$E+06$	1.34$E+07$	1.38$E+08$	3.41$E+04$	4.66$E+04$	1.05$E+04$	1.97$E+04$	1.25$E+04$
	Ave	**6.76E+03**	7.64$E+07$	1.60$E+07$	1.49$E+07$	1.54$E+08$	1.25$E+06$	5.65$E+04$	1.06$E+04$	1.72$E+04$	1.68$E+04$
	Std	**1.69E+03**	1.76$E+07$	1.35$E+07$	1.22$E+07$	9.75$E+07$	2.45$E+06$	5.64$E+04$	3.59$E+03$	6.03$E+03$	2.37$E+04$
	Rank	1	9	8	7	10	6	5	2	4	3
Average rank		3.07	7.34	6.00	7.41	9.20	6.66	4.90	3.31	4.35	2.80

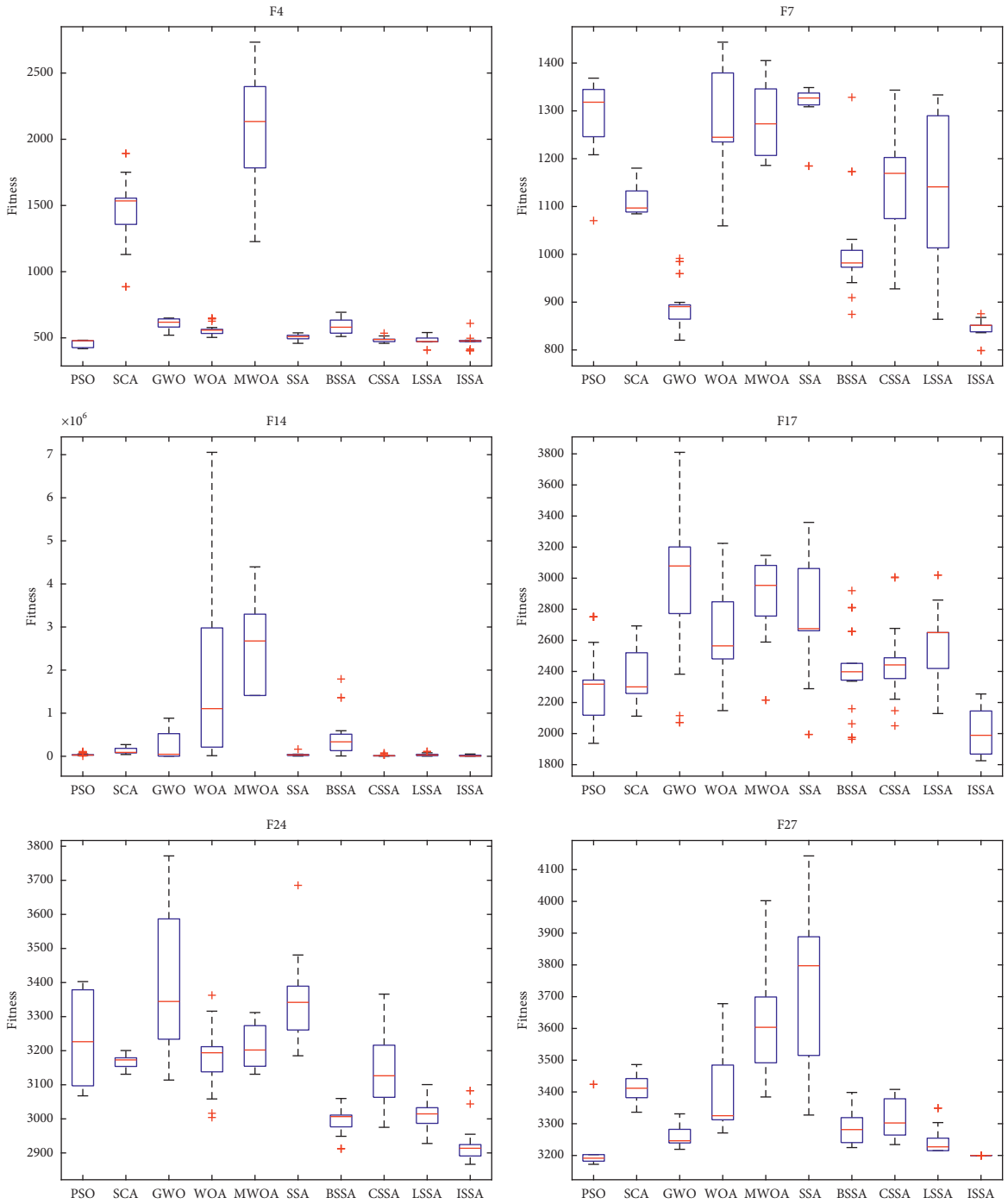


Figure 9: Box diagram of each algorithm.

speed and accuracy are better than SSA. In the sinusoidal input response, ISSA can almost coincide with rin, and the tracking effect is significant. The tracking effect of SSA is slightly inferior. At the same time, the convergence speed and accuracy of ISSA are slightly better than SSA. The above results verify that ISSA has good algorithm performance, can quickly and accurately complete the PID parameter tuning, and help the system have shorter response time, higher system control accuracy, and better robustness. So far, the practicability has been proved.

7. Robot Path Planning

In PID parameter tuning, the dimension of the practical application is low. Therefore, this paper selects the discrete problem of more complex path planning to further verify the practicability of ISSA and makes a comparative experiment with SSA. In path planning, each sparrow is a feasible path. The environment modeling adopts the grid method, and the obstacles at the equivalent position are calculated according to the grid value. Grid number 0 is defined as feasible area and 1 as obstacle area.

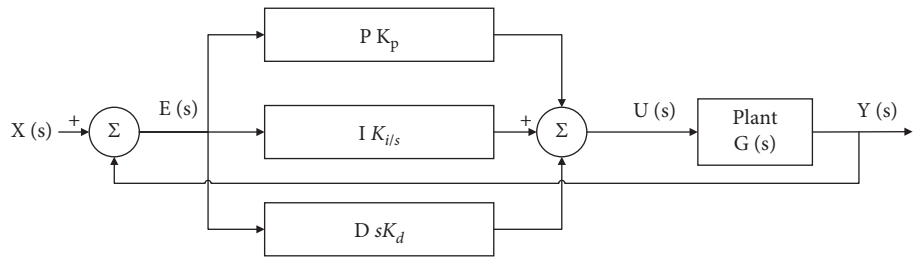


Figure 10: PID controller structure diagram.

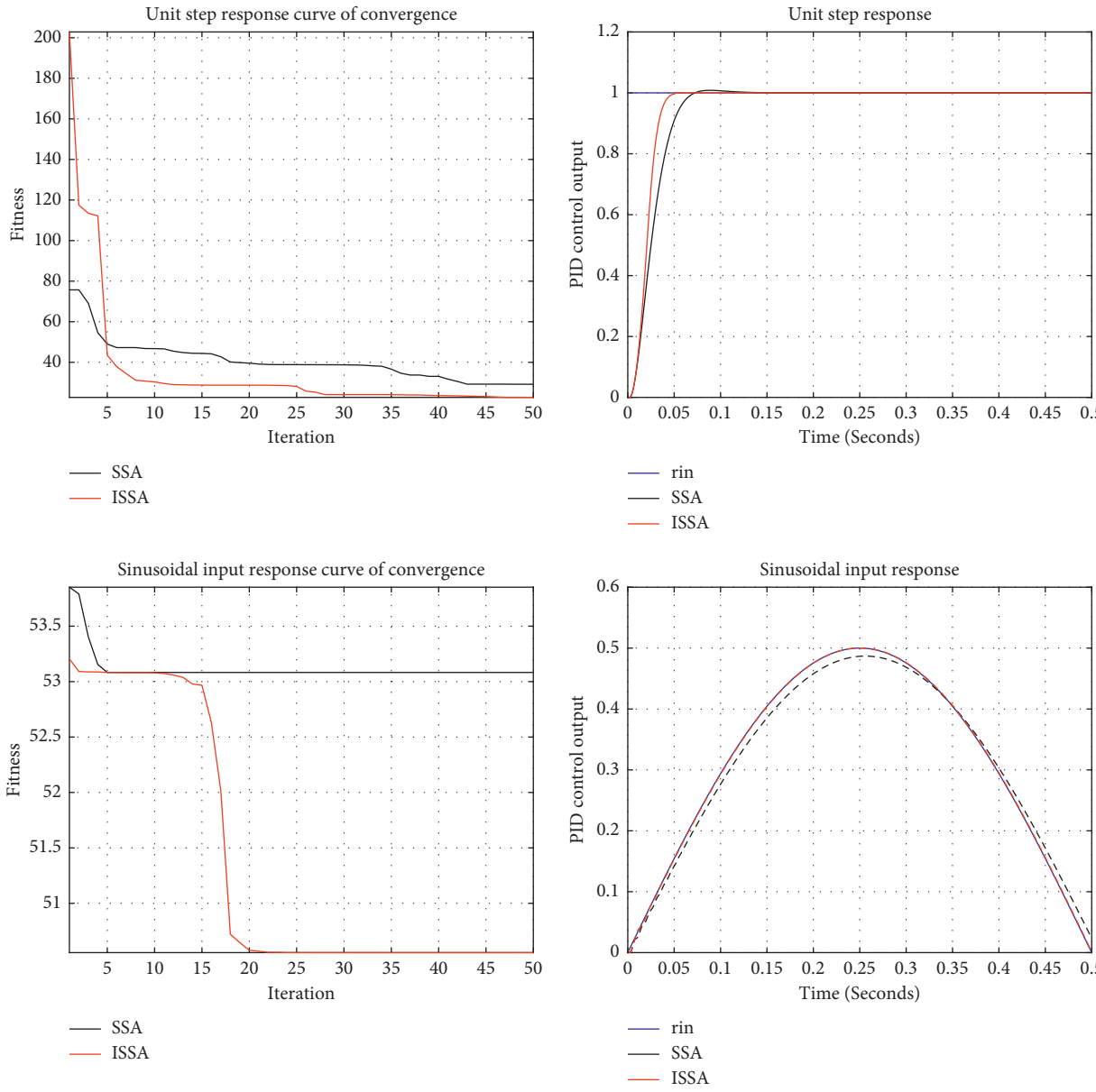


Figure 11: Convergence curve and PID control output.

Then, the robot can plan the path on the grid specified as 0, and dimension D is the column number of the grid map. The cost function of the path length of the i-th sparrow is shown in

$$f(x) = \sum_{j=1}^{D-1} \sqrt{\left(x_{j+1} + x_j\right)^2 + \left(y_{j+1} - y_j\right)^2}. \quad (31)$$

Table 9: PID parameter setting results.

Algorithm	Unit step				Sinusoidal input			
	Fitness	K_p	K_i	K_d	Fitness	K_p	K_i	K_d
SSA	29.1777	10	0.221046	0.126894	53.0827	10	10	1.90441
ISSA	**22.7045**	41.4593	0.596364	0.402723	**50.5554**	286.6486	21.19153	0.898048

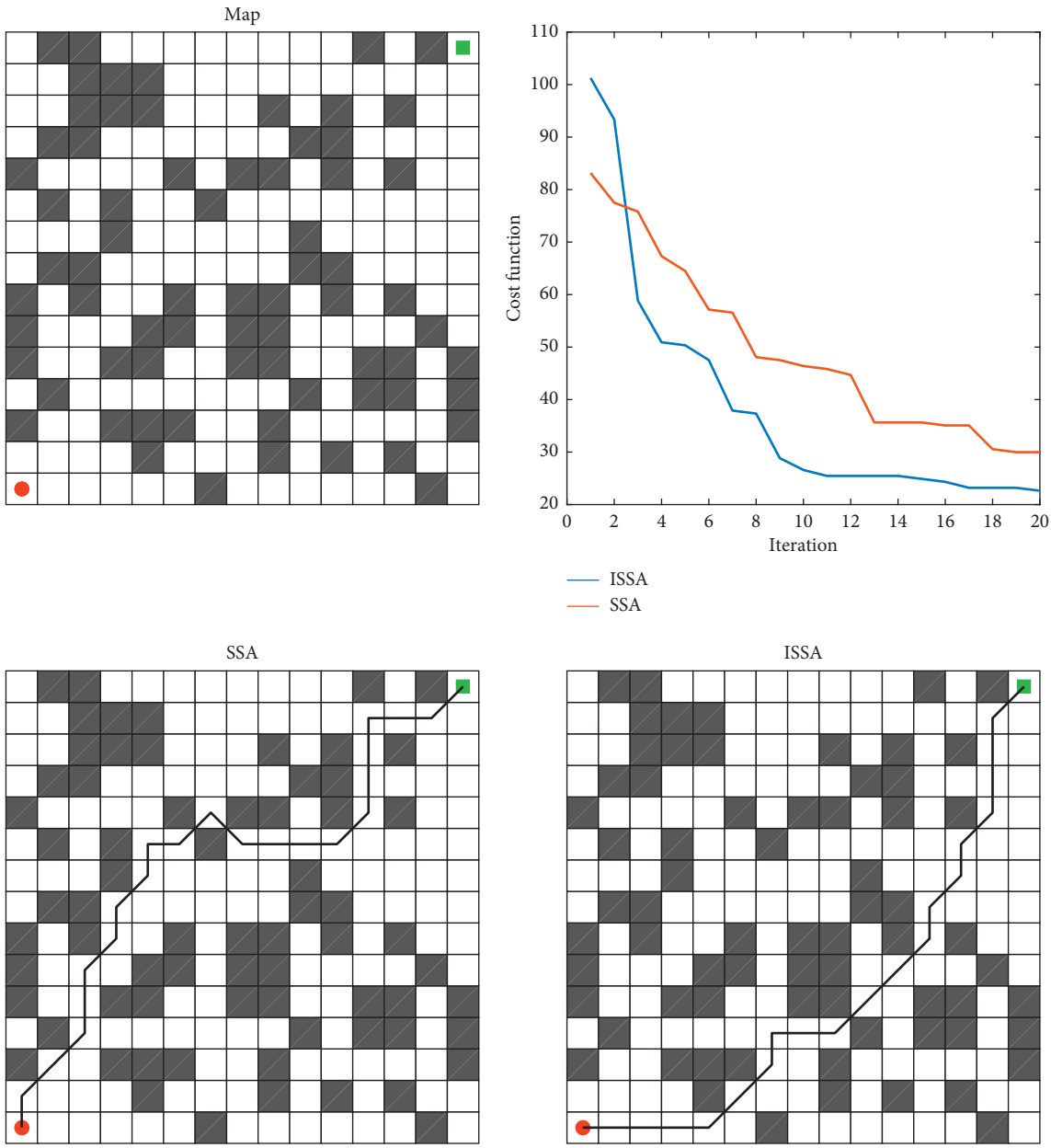


Figure 12: Convergence curve and optimal path.

In equation (31), j is the j-th dimension of a sparrow. Set each algorithm at 15×15, and the best path is shown in Figure 12. In order to eliminate the contingency, each algorithm is tested for 10 times, the optimal value, worst value, average value, and standard deviation of the fitness value of each algorithm are calculated, and these four indexes are used to measure the stability and feasibility of each algorithm. The optimization statistics of each algorithm are shown in Table 10.

It can be seen from the Table 10 and Figure 12, the minimum cost of ISSA planning is 19.7990, while the minimum cost of SSA is 22.6274. It can be seen that ISSA has

Table 10: Robot path planning results.

Algorithm	Fitness			
	Best	Worst	Ave	Std
SSA	22.6274	39.5980	29.9813	6.5115
ISSA	**19.7990**	**28.2843**	**22.6274**	**3.4641**

strong path planning ability. According to other indicators, it can be seen that the path planned by ISSA has good stability. Therefore, ISSA has a good effect in more complex robot path planning and can plan a more stable and safer path.

8. Conclusions

This paper proposes an improved sparrow search algorithm based on iterative local search strategy, which introduces four strategies: variable helix factor, improved iterative local search, the lens imaging with changing focusing ability, and improved boundary control. ISSA overcomes the shortcomings of poor utilization of current individuals and lack of effective search and effectively improves the problems of falling into local optimal solution and low optimization accuracy. The test function results show that ISSA has good optimization performance and universality. The results of PID parameter tuning and robot path planning show that ISSA algorithm has good practicability.

The improved ISSA has good optimization performance, but it also has some shortcomings. For example, it can only find the optimal value in some functions, and other performance indicators are poor and unstable; a certain amount of work is added, resulting in a longer consumption time of the algorithm; and it did not improve the search scope of discoverers. In view of the shortcomings, we still need to do some work in the future: first, how to improve the stability of the algorithm; second, how to improve the search ability of followers; third, how to balance the time and optimization ability of the algorithm; and fourth, how to improve the search ability of discoverers on the basis of discoverers.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by the regional foundation of the National Natural Science Foundation of China (No. 61703411).

References

[1] W. L. Liu, Y. J. Gong, W. N. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2019.

[2] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, pp. 1430–1442, 2019.

[3] F. Zhao, R. Ma, and L. Wang, "A self-learning discrete jaya algorithm for multiobjective energy-efficient distributed no-idle flow-shop scheduling problem in heterogeneous factory system," *IEEE Transactions on Cybernetics*, 2021.

[4] F. Zhao, X. He, and L. Wang, "A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem," *IEEE Transactions on Cybernetics*, vol. 51, pp. 5291–5303, 2020.

[5] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the 2009 International Symposium on Stochastic Algorithms*, Sapporo, Japan, 2009.

[6] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.

[7] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[8] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[9] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, 2016.

[10] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.

[11] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, and A.-I. Szedlak-Stinean, "Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1042–1052, 2021.

[12] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.

[13] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, Article ID 115079, 2021.

[14] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionic Engineering*, vol. 18, no. 3, pp. 674–710, 2021.

[15] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[17] 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*, vol. 5, Orlando, FL, USA, 1997.

[18] C. R. Houck, J. Joines, and M. G. Kay, *A Genetic Algorithm for Function Optimization: A Matlab Implementation*, NCSU, Raleigh, NC, USA, 1995.

[19] Y. Li, S. Wang, Q. Chen, and X. Wang, "Comparative study on some new swarm intelligence optimization algorithms," *Computer Engineering and Application*, vol. 56, no. 22, pp. 1–12, 2020.

[20] X. Lv, X. Mu, and J. Zhang, "Multi-threshold image segmentation based on improved sparrow search algorithm," *Systems Engineering and Electronics*, vol. 43, no. 2, pp. 318–327, 2021.

[21] X. Lv, X. Mu, J. Zhang, and Z. Wang, "Chaos sparrow search optimization algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 8, pp. 1712–1720, 2021.

[22] A. Tang, T. Han, D. Xu, and L. Xie, "UAV track planning method based on chaotic sparrow search algorithm," *Computer Applications*, vol. 41, no. 7, pp. 2128–2136, 2021.

[23] C. Ouyang, D. Zhu, and F. Wang, "A learning sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 3946958, 23 pages, 2021.

[24] D. Zhang, H. Xu, Y. Wang, T. Song, and L. Wang, "Whale optimization algorithm embedded in circle mapping and dimensional keyhole imaging reverse learning," *Control and Decision Making*, vol. 36, no. 5, pp. 1173–1180, 2021.

[25] G. Wu, R. Mallipeddi, and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization*, National University of Defense Technology, Changsha, China, 2017.

[26] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Technical report, Zhengzhou University, Zhengzhou, China, 2016.

[27] Y. Chen, Y. Ma, and W. Yun, "Application of improved genetic algorithm in PID controller parameters optimization," *Telkomnika*, vol. 11, no. 3, pp. 1524–1530, 2013.

[28] C. Gan, W. Cao, M. Wu, and X. Chen, "A new bat algorithm based on iterative local search and stochastic inertia weight," *Expert Systems with Applications*, vol. 104, pp. 202–212, 2018.

[29] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, Berlin, Germany, pp. 65–74, 2010.

[30] D. Molina, A. LaTorre, and F. Herrera, "SHADE with iterative local search for large-scale global optimization," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018.

[31] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 392–403, 1998.

[32] X. Xia, J. Liu, and Y. Li, "Particle swarm optimization algorithm with reverse-learning and local-learning behavior," *Journal of Software*, vol. 9, no. 2, pp. 350–357, 2014.

[33] J. He, Z. Peng, D. Cui, and Q. Li, "A teaching and learning optimization algorithm based on multi-inverse learning," *Engineering Science and Technology*, vol. 51, no. 6, pp. 159–167, 2019.

[34] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[35] L. Wen, T. Wu, M. Tang, M. Xu, and S. Cai, "Gray wolf optimization algorithm based on lens imaging learning strategy," *Acta Automatica Sinica*, vol. 46, no. 10, pp. 2148–2164, 2020.

[36] C. Ouyang, D. Zhu, and Y. Qiu, "Lens learning sparrow search algorithm," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9935090, 17 pages, 2021.

[37] S. Peng, Z. Wu, X. Zhou, and C. Deng, "Improved particle swarm algorithm based on the reverse learning model of refraction principle," *Journal of Electronics*, vol. 43, no. 11, pp. 2137–2144, 2015.

[38] Q. Liang, B. Chen, H. Wu, C. Ma, and S. Li, "A novel modified sparrow search algorithm with application in side lobe level reduction of linear antenna array," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9915420, 25 pages, 2021.

[39] S. He, Z. Cao, and S. Yu, "PID parameter optimization based on pollination algorithm," *Computer Engineering and Applications*, vol. 52, no. 17, pp. 59–62, 2016.