

## Research Article

# Prediction of Train Arrival Delay Using Hybrid ELM-PSO Approach

Xu Bao <sup>1</sup>, Yanqiu Li <sup>2</sup>, Jianmin Li <sup>2</sup>, Rui Shi <sup>2</sup> and Xin Ding <sup>2</sup>

<sup>1</sup>College of Traffic Engineering, Huaiyin Institute of Technology, Jiangsu 223001, China

<sup>2</sup>School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Xu Bao; [baoxu@hyit.edu.cn](mailto:baoxu@hyit.edu.cn)

Received 10 May 2021; Accepted 4 June 2021; Published 14 June 2021

Academic Editor: Linchao Li

Copyright © 2021 Xu Bao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this study, a hybrid method combining extreme learning machine (ELM) and particle swarm optimization (PSO) is proposed to forecast train arrival delays that can be used for later delay management and timetable optimization. First, nine characteristics (e.g., buffer time, the train number, and station code) associated with train arrival delays are chosen and analyzed using extra trees classifier. Next, an ELM with one hidden layer is developed to predict train arrival delays by considering these characteristics mentioned before as input features. Furthermore, the PSO algorithm is chosen to optimize the hyperparameter of the ELM compared to Bayesian optimization and genetic algorithm solving the arduousness problem of manual regulating. Finally, a case is studied to confirm the advantage of the proposed model. Contrasted to four baseline models (k-nearest neighbor, categorical boosting, Lasso, and gradient boosting decision tree) across different metrics, the proposed model is demonstrated to be proficient and achieve the highest prediction accuracy. In addition, through a detailed analysis of the prediction error, it is found that our model possesses good robustness and correctness.

## 1. Introduction

With the rapid development of society and the continuous improvement of people's quality of life, people have put forward higher requirements for the reliability and punctuality of high-speed railway transportation [1]. However, the train will inevitably be disturbed by a large number of random factors in the process of running, which will lead to the train delay. For one thing, train delay will change the structure of train diagram, increase the cost of railway operation and the difficulty of reasonable utilization of transportation resources, and have a great negative impact on the reliability and punctuality of high-speed railway operation. For another, it will increase the travel time of passengers, affect their travel plans, and bring serious inconvenience to passengers [2]. Therefore, accurate forecast of train delay is of great significance for high-speed train operation organization, transportation service quality improvement, and operation safety [3].

The traditional models are a classical approach for train delay prediction, such as probability distribution models [4, 5], regression models, event-driven methods, and graph theory-based approaches. For the probability distribution model, Higgins and Kozan proposed an exponential distribution model, which applied a three-way, two-block station train delay propagation signal system, to estimate delays of trains caused by train operational accidents [4]. Through the assessment of the linear relationship between several independent features and dependent features [2], regression models were widely employed to predict train delays, dwell times, and running times [6, 7]. However, the main drawback of regression models is that the ability of linear analytic model relies much on internal and mathematical assumptions. They are good at capturing the linear relationship between features and dealing with low-dimensional data, not simple linear data, such as train operation data [8]. For event-driven and graph theory-based methods, Kecman and Goverde [9] used timed event graph with dynamic weight to predict train running time. Milinković

et al. [10] used fuzzy Petri net to predict the train delays; the model considers the characteristics of the hierarchical structure and fuzzy reasoning to simulate the train operation and predicts the train delays in different delay scenarios. Huang et al. [8] used graph theory to calculate the degree and propagation range of train delay under specific condition. Although the traditional model started early in the study of delay prediction, it generally has the limitation of poor generalization performance and is only suitable for specific scenarios.

Recently, the application of machine learning methods to predict train delays has been widely concerned by researchers, which makes up for the shortcomings of traditional methods [2]. The purpose of Peters et al. was to utilize the historical travel time between stations to predict the train arrival time more precisely [11]. The moving average algorithm of historical travel time and KNNs of last arrival time algorithm were employed and estimated. Some researchers are devoted to ANNs to predict train delays [12–14]. The aim of [14] was to propose preminent ANNs to predict the train delay of Iran railway with three different models, including standard real number, binary coding, and binary set encoding inputs. Nevertheless, the prediction accuracy of ANNs cannot meet the needs of actual delay management, and the parameter adjustment is complex. Marković et al. [2] proposed a support vector regression model in train delay problem of passenger train, which captured the relationship between the arrival delay and a variety of changing external factors, and compared it with the artificial neural networks. The results indicated that the support vector regression method outperformed the ANNs. Another neural network has been proposed in recent years. A Bayesian network model for predicting the propagation of train delays was presented by [15]. In view of the complexity and dependence, three different BN schemes for train delay prediction were proposed, namely, heuristic hill-climbing, primitive linear, and hybrid structures [16]. The results turned out to be quite satisfying. Recently, it has become popular to combine several models to capture various characteristics of train operation data to predict train delay. A study developed a train delay prediction model, which combines convolutional neural networks, long short-term memory network, and fully-connected neural network architectures to solve this issue [17].

To improve the backpropagation algorithm and simplify the setting of learning parameters of general machine learning models, the ELM algorithm was proposed by Bin Huang et al. [18]. ELM has the advantages of small computation, good generalization, and fast convergence. On account of these advantages, ELM has been frequently applied to regression problems in the real world [19–22]. Therefore, a new study that combined a shallow ELM and a deep ELM tuned via the threshold out technique was employed to predict train delays, taking the weather data into account [23].

Parameter adjustment is another critical factor to guarantee the good performance of machine learning models [24, 25]. Although the well-known random search algorithm can achieve the purpose of optimization, it generates all the solutions randomly without considering the

previous solutions. An adjusting parameter model, called PSO, has become one of the widely used parameter adjustment methods because of its ability to address intractable matters in the real world. Only the optimal particle of PSO transmits the information to the next particle in the iterative evolution process. As a consequence, the searching speed of PSO is faster than random search and grid search [26]. The experiment [27] did just prove the advantage of PSO. By comparing the performance of PSO with random search algorithm for the optimal control problem, [27] found out that PSO was capable of locating better solution with the same number of fitness function calculations than random search algorithm.

Therefore, according to what the author has learnt, we propose PSO to optimize the hyperparameter of ELM to forecast train arrival delays.

The contributions this paper makes are as follows:

- (1) The main features affecting the train delay prediction are evaluated by the extra trees classifier. Then, the proposed model is constructed based on these features which possess spatiotemporal characteristics (train delays at each station). In this way, the interpretability of the proposed model is improved.
- (2) The proposed model is applied to the arrival delay prediction of trains on HSR line, which suggests a brand-new perspective for the train delay prediction problem. In addition to solving the drawbacks of backpropagation algorithm, the advantage of ELM-PSO is also to solve the arduous problem of manual regulating the hidden neurons of ELM better than random search and Bayesian optimization at accuracy and efficiency.
- (3) We perform experiments on a section of the Wuhan-Guangzhou (W-G) HSR line. The proposed model not only is compared to other two adjusting parameter models, but also is contrasted with four prediction models from different perspectives. Our model turns out to have an extraordinary ability in managing large-scale data in accuracy.

The remainder of this paper is distributed as follows: in Section 2, the train delay problem and selection of characteristic features are described. The hybrid ELM-PSO approach is introduced in detail in Section 3. The data description and experimental settings are presented in Section 4. The performance analysis is discussed in Section 5. Finally, conclusions are presented in Section 6.

## 2. Description of the Train Delay Problem

Train delay problem is visualized in Figure 1 to assist in comprehending this abstract problem. The train delay contains two contents, train arrival delay and train departure delay. For a station  $s_n$ ,  $t_{Asn}$  represents the time that one train is scheduled to arrive at station  $s_n$  and the same goes for  $t_{Dsn}$ , which implies the time that one train is scheduled to depart at station  $s_n$ . Certainly, the train will have its own actual timetable due to changing external factors, which are

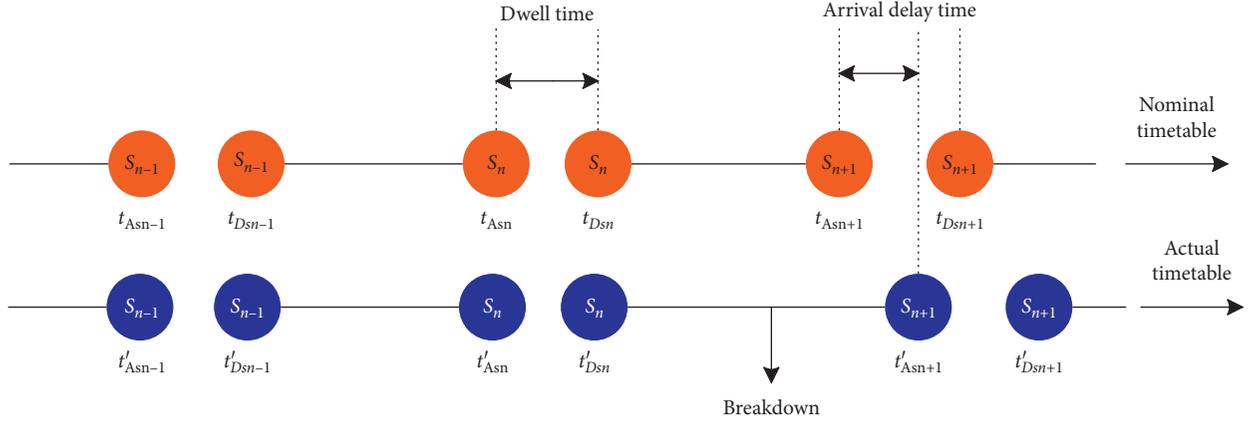


FIGURE 1: Conversion from the train itinerary to mathematical notation.

expressed as  $t'_{Asn}$  and  $t'_{Dsn}$ , respectively. The difference between the actual and scheduled arrival time at station  $s_n$ ,  $t'_{Asn} - t_{Asn}$ , is referred to as the train arrival delay. The same goes for the train departure delay  $t'_{Dsn} - t_{Dsn}$ . This is the primitive description of the train delay problem.

This paper only focuses on the train arrival delay prediction. We suppose that there is an aimed train  $T_k$ , which is at present station  $s_n$  at time  $t_{Asn}$ . Our purpose is to predict the arrival delay ( $t'_{Asn+1} - t_{Asn+1}$ ) of the targeted train  $T_k$  at its following station  $s_{n+1}$  for all conditions according to the information of train  $k$  at stations  $s_n$ ,  $s_{n+1}$ , and  $s_{n-1}$ , which is made up of the following nine features:

- (1) The station code ( $X_1$ )
- (2) The train number ( $X_2$ ), which indicates the number of the trains
- (3) The length between the present station and the next station ( $D_{sn+1} - D_{sn}$ ) ( $X_3$ )
- (4) The scheduled running times between the present station and the previous station ( $t_{Asn} - t_{Asn-1}$ ) ( $X_4$ )
- (5) The actual running times between the present station and the previous station ( $t'_{Asn} - t'_{Asn-1}$ ) ( $X_5$ )
- (6) The scheduled running times between the present station and the next station ( $t_{Asn+1} - t_{Asn}$ ) ( $X_6$ )
- (7) The actual running times between the present station and the next station ( $t'_{Asn+1} - t'_{Asn}$ ) ( $X_7$ )
- (8) Buffer time, which indicates the difference between  $X_6$  and actual minimum running time of all trains between the present station and the next station ( $X_6 - \min\{T_1(t'_{Asn+1} - t'_{Asn}) \dots T_k(t'_{Asn+1} - t'_{Asn})\}$ ) ( $X_8$ )
- (9) The arrival delay time at the present station ( $t'_{Asn} - t_{Asn}$ ) ( $X_9$ )

$Y$  represents the arrival delay time at the next station  $s_{n+1}$  of train  $T_k$  ( $t'_{Asn+1} - t_{Asn+1}$ ).

There are multiple potential interdependent features (e.g., the train number, the length between two adjacent stations) that are intently related to train delay prediction.

Based on the collected data and the experience of dispatchers, we ultimately select nine features that are possible to influence train delays.

We apply extra trees classifier to analyze the correlation between all features and train delays. The results are exhibited in Figure 2. As shown in the figure, the deeper the red, the higher the importance. There is no doubt that  $X_9$  has the highest importance with  $Y$ . The actual and scheduled running times between the present station and the next station also contribute largely to the accurate prediction of  $Y$ . Moreover, the buffer time, which is an important factor affecting the length of the train recovery time, is also comparatively prominent in delay prediction process. Taking the buffer time into account allows us to obtain more realistic prediction results.

The train arrival delay prediction problem in this paper is transformed into the following expression:

$$Y = f(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9), \quad (1)$$

where  $X_i$  is the information of train  $T_k$  running through stations  $s_n$ ,  $s_{n+1}$ , and  $s_{n-1}$ ,  $Y$  is the arrival delay time at the following station  $s_{n+1}$  of train  $T_k$ , and  $f(x)$  is the prediction process.

### 3. Methodology

This paper proposes a hybrid model of ELM and PSO for train delay prediction. ELM is widely used in regression problems because of its advantages of small computation, good generalization performance, and fast convergence speed [19–21]. PSO algorithm is a random and parallel optimization algorithm, which has the advantages of fast convergence speed and simple algorithm [25, 28]. Therefore, we aim to combine the advantages of ELM and PSO algorithm to improve the behavioral knowledge in the delay prediction domain. For the principle of ELM and PSO, one can refer to Li et al. [29], Perceptron et al. [30], and Zhang et al. [31]. The running process of the proposed hybrid method is as follows:

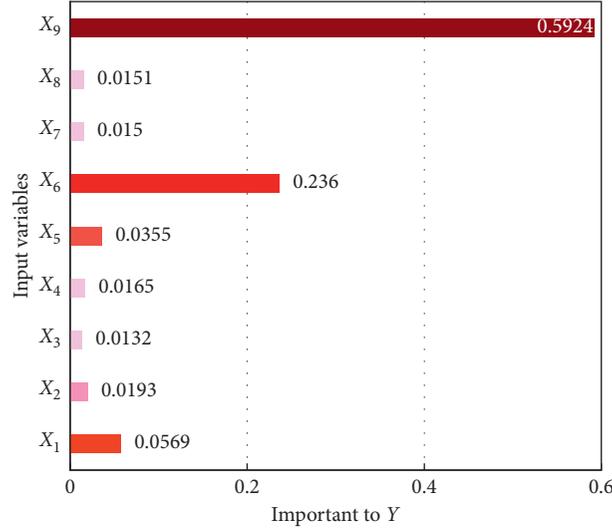


FIGURE 2: Bar chart of the correlations between the nine input features and output  $Y$ .

Step 1: data preprocessing. First, 9 features mentioned in Section 2 are generated a  $N \times 9$  matrix, where  $N$  represents the total number of events according to the train operation records. Second, remove abnormal delay (trains may be canceled due to some emergencies) to reduce its interference with predictions. Third, fill in the missing data according to the adjacent data around the missing ones.

Step 2: initializing the parameters and population. Parameters such as maximal iteration number, population size, and speed and position of the first particle are initialized. Each particle  $l$  has its own position  $A_l(ite)$  and speed  $V_l(ite)$ . The position of each particle in the population is equivalent to the number of neurons in the hidden layer of ELM. Therefore, there is merely one dimension of each particle:

$$A_l(ite) = h_l(ite), \quad l = 1, \dots, k, \quad (2)$$

where  $h_l(ite)$  represents the number of hidden layer neurons of ELM in the  $ite_{th}$  iteration.

Step 3: ELM (hidden layer activation function: sigmoid function) is used. The processed feature set  $X$  and the position of particles (the number of hidden layer neurons) generated by PSO are input into ELM. Consequently, ELM can output the weight matrix under the current number of hidden layer neurons. The function of calculating the fitness of particles is as follows:

$$p_l(ite) = \text{fitness}(A_l(ite), X) = \sqrt{\frac{\sum_{n=1}^N (y_n - f_n(x))^2}{N}},$$

$$A_l(ite) = h_l(ite), \quad l = 1, \dots, k, \quad (3)$$

where  $N$  is the number of samples,  $y_n$  is the actual output value on test set, and  $f_n(x)$  is the predicted output value on test set.

Step 4: calculate the fitness of each particle, and compare to update the current best fitness and its particle location.

Step 5: start the iteration. PSO will update the positions and velocities of all particles, and then repeat step 4. If the maximum number of iterations is exceeded, it will end the process.

Step 6: output the results. We can obtain the output value on test set as well as the optimal number of hidden layer neurons.

The specific flowchart is shown in Figure 3.

## 4. Application to a Case Study

**4.1. Dataset Description.** The data employed to verify the ELM-PSO are obtained from the dispatching office of a railway bureau. The 15 stations applied in the study include a section, the length of which is 1096 km from CBN to GZS on the double-track W-G HSR line. There are more than 400,000 data points used in this study, with a time span from October 2018 to April 2019. The train original operation data and route map of the targeted 15 stations on the W-G HSR line are shown in Table 1 and Figure 4.

Analysis of the delay ratio of each station reveals not only the condition of each station but also an increasing emphasis on the indispensability of train arrival delay prediction, which contributes to improving the ability of each station to cope with and even inhibit the increase in train arrival delays. Trains with arrival delay greater than 4 minutes are considered as delayed trains. What is intuitively presented in Figure 5 is that the delay ratios of all the stations are basically

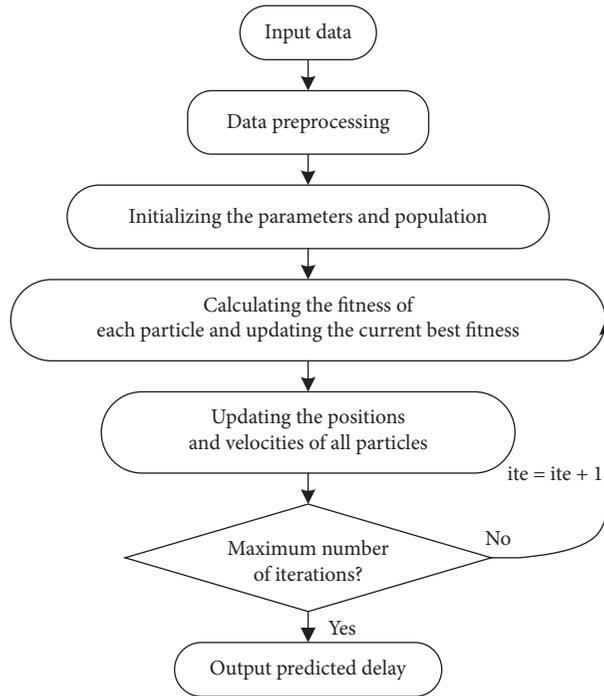


FIGURE 3: The flowchart of ELM-PSO method.

TABLE 1: Train operation data format in the database.

Station	Station code	Date	Actual arrival	Actual departure	Train	Scheduled arrival	Scheduled departure
GZS	369	2018/7/27	12:04	12:04	G100	12:05	12:05
GZN	368	2018/7/27	12:19	12:19	G100	12:19	12:19
QY	367	2018/7/27	12:26	12:26	G100	12:26	12:26
YDW	366	2018/7/27	12:37	12:37	G100	12:38	12:38

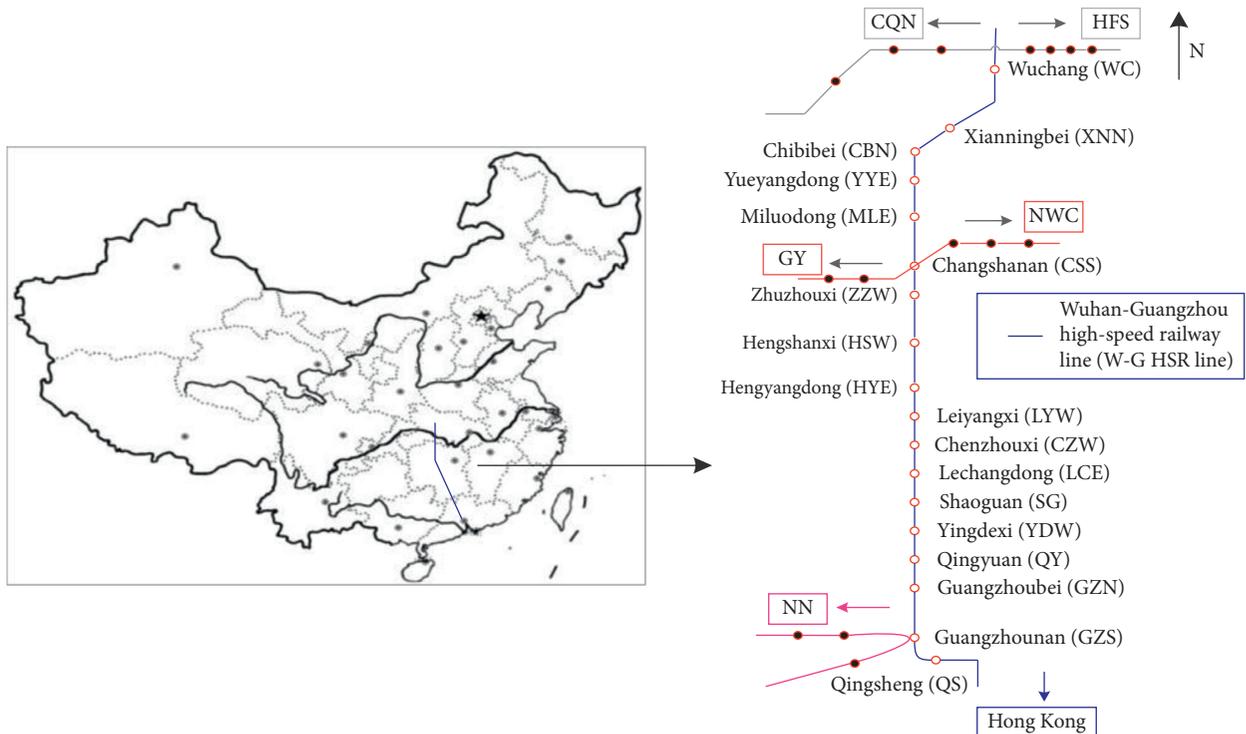


FIGURE 4: Map of the W-G HSR line.

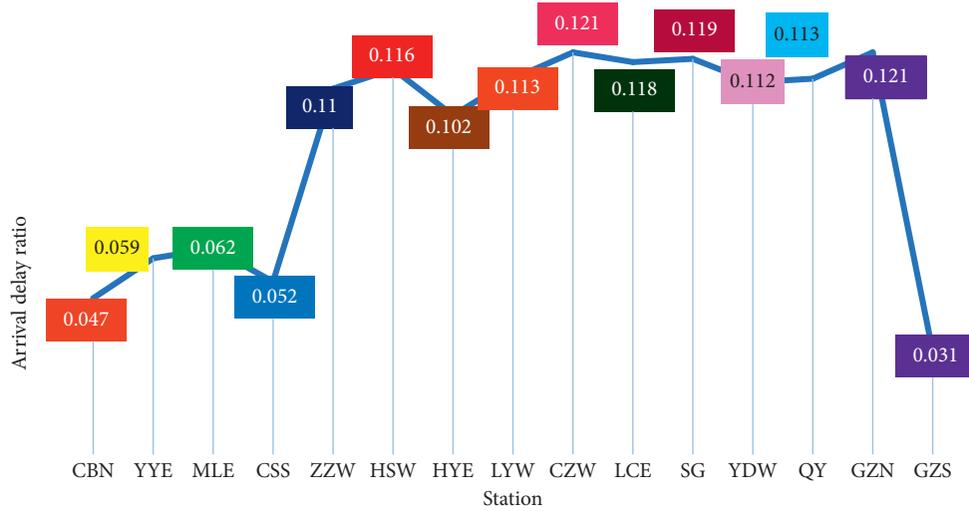


FIGURE 5: Arrival delay ratio for each station.

not optimistic. At the same time, the delay ratios of the two targeted stations, CZW and GZN, are particularly dreadful, with arrival delay ratios of 0.12. Our goal is to minimize the arrival delay ratio by predicting the arrival delay at each station.

#### 4.2. Experimental Settings

**4.2.1. Baseline Models.** In order to compare the performance of our proposed method, the k-nearest neighbor (KNN), categorical boosting (CB), gradient boosting decision tree (GBDT), and Lasso are used as baseline models. We take 20% of the dataset as the test set and the rest as the training set. The experiment runs in Python in an environment with an Intel® Core i5-6200U processor 2.13 GHz and 8 GB RAM. Briefly, an overview description and hyperparameter settings of each model are as follows:

- (1) KNN: KNN algorithm is extensively applied in differing applications massively, owing to its simplicity, comprehensibility, and relatively promising manifestation [32].

- ① N\_neighbors = 15
- ② Weights = uniform
- ③ Leaf\_size = 30
- ④ P = 2

- (2) CB: CB is a machine learning model based on gradient boosting decision tree (GBDT) [33, 34]. CB is an outstanding technology, especially for datasets with heterogeneous features, noisy data, and complex dependencies.

- ① Depth = 3
- ② Learning\_rate = 0.1
- ③ Loss\_function = RMSE

- (3) GBDT: GBDT has been employed to numerous problems [35], which has many nonlinear transformations and strong expression ability and does

not need to do complex feature engineering and feature transformation.

- ① N\_estimators = 30
- ② Loss = ls
- ③ Learning\_rate = 0.1

- (4) Lasso: Lasso is a prevailing technique, capable of simultaneously performing regularization and feature filtering. Furthermore, data can be analyzed from multiple dimensions by Lasso [36].

- ① Alpha = 3.0.
- ② Max\_iter = 1000.
- ③ Selection = cyclic.

**4.2.2. Evaluation Metrics.** Root mean squared error (RMSE), mean absolute error (MAE), and R-squared are selected to assess the models. The definitions of the error metrics are shown in equation (4), equation (5), and equation (6):

$$\text{MAE} = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{N}, \quad (4)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}}, \quad (5)$$

$$R\text{-squared} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2}, \quad (6)$$

where  $y_i$  is an observed value,  $\hat{y}_i$  is a predicted value,  $\bar{y}_i$  is the average value of  $y_i$ , and  $N$  represents the sample size.

**4.2.3. Hyperparameter Tuning Models.** We compare PSO with the other two hyperparameter tuning models to ascertain the most satisfying one. The overview and hyperparameter settings of each model are as follows:

- (1) PSO: to locate the optimal hyperparameter of the ELM, the parameter settings of the PSO algorithm are as follows. PSO has 20 particles at each iteration, and there are altogether 20 iterations, which is equivalent to 400 iterations of Bayesian optimization.
  - ① Number of particles = 20
  - ② Fitness function: RMSE on test set
  - ③ Search dimension = 1
  - ④ Particle search range = [1, 2000]
  - ⑤ Maximum number of iterations = 20
- (2) BO (Bayesian optimization): BO calculates the posterior probability distribution of the first  $n$  points through a substitution function and obtains the objective function of each hyperparameter at each value point.
  - ① Objective function: RMSE on test set
  - ② Substitution function: Gaussian process regression
  - ③ Acquisition function = UCB (upper confidence bound)
  - ④ Hyperparameter search range = [1, 2000]
  - ⑤ Maximum number of iterations = 400
- (3) GA (genetic algorithm): the traditional iterative model is easy to fall into the trap of local minima, which makes the iteration impossible to continue. GA overcomes the phenomenon of “dead loop” and is a global optimization algorithm [37].
  - ① Objective function: RMSE on test set
  - ② Hyperparameter search range = [1, 2000]
  - ③ Generations = 20
  - ④ Population size = 20
  - ⑤ Maximum number of iterations = 400

## 5. Performance Analysis

**5.1. PSO Optimization Result Comparison.** The process of PSO tuning the hyperparameter is shown in Figure 6. The fitness value achieves minimum after five iterations. The best fitness value is 1.0387 on test set when there are 1462 neurons of the ELM. The structure of the network is optimal correspondingly.

The search range [1–2000] of hyperparameter is determined by manually trying several values in the range of [1–10000]. When the hyperparameter value is greater than 2000, the fitness tends to be stable. Also, the time consumption is multiplied acutely. Ultimately, we decide to limit the search range to [1–2000], weighing time consumption and precision.

The computational cost is shown in Table 2, and the results are the optimal results of each model running several times. We gain two observations from this table. First, the optimal particle number of ELM-PSO always focuses on 1462; the only difference is the number of iterations at best RMSE. Second, compared with ELM-BO and ELM-GA,

ELM-PSO is the ideal model that takes the shortest time to locate the optimal fitness on the test set.

**5.2. Model Accuracy Comparison.** In this section, the performance comparison between ELM-PSO and baseline models is performed.

First, we compare the overall performance of the five models. The evaluation metrics are  $R$ -squared, the MAE, and the RMSE. The corresponding results on test set and training set are summarized in Tables 3 and 4, respectively. The ELM-PSO model performs optimally among the five models in not only the training set ( $R$ -squared = 0.9973; MAE = 0.3377; RMSE = 0.8247), but also the test set ( $R$ -squared = 0.9955; MAE = 0.3490; RMSE = 1.0387). Although the running time of our model has no obvious advantage compared with other models on test set, it is within the tolerable range. Also, we notice that there are models that perform well in the training set, but are not good in the test set, which reveals the paramountcy of enough generalization ability of models in the prediction problem.

Then, by separating the delay duration into three bins (i.e., [0–1200 s], >1200 s, and all delayed trains (trains with arrival delay greater than 240 seconds)), we attempt to measure the capability of the benchmark models and our model to seize the features of train delays to varying degrees on test set. As is distinctly shown in Table 5, the proposed model outperforms the other benchmark models in each time horizon and each evaluation metric, achieving, for example, an RMSE of 0.5201 in the first bin. This finding is taken as evidence that our model can constantly adjust itself to capture the characteristics of varying degrees of train delays to enhance the prediction accuracy. To further assess the performance of our model, the comprehensive analyses are discussed in a later section.

**5.3. Further Analysis.** On the basis of the previous section, we will evaluate the performance of the ELM-PSO model from other angles, including the prediction errors for each station precisely, the prediction correctness, and the robustness.

First and foremost, the errors of the ELM-PSO model for the predicted arrival delays are calculated at the station level on test set. Viewing the overall situation in Figure 7, we have noticed that the prediction errors are low. The MAE and  $R$ -squared both remain stable at each station. And the RMSEs for different stations are mostly less than 90s. However, great fluctuations occur at the YYE and QY stations. Reasons resulting in such phenomena are that the two stations are close to the transfer stations and the buffer times of YYE and QY are both small. The prediction accuracy at these stations tends to be slightly hindered by these factors.

In addition, to put forward more detailed and embedded results, we describe the correctness of the absolute residual between the predicted values and the actual values for each station from three intervals (i.e., <30 s, 30 s–60 s, and 60 s–90 s) (Figure 8). In <30 s interval, the correctness of

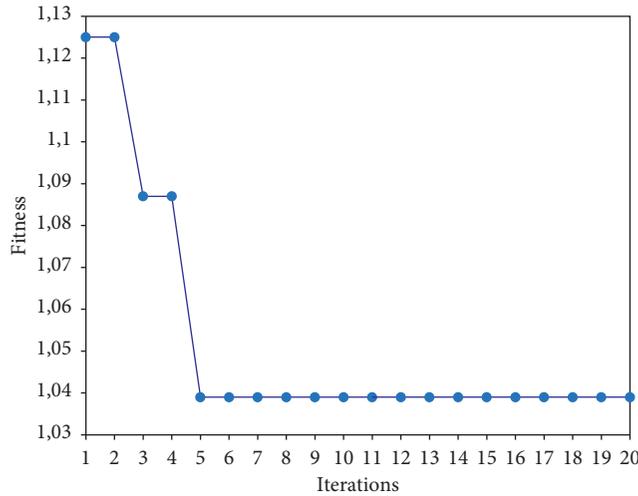


FIGURE 6: Convergence discriminant graph of PSO optimization.

TABLE 2: The performance of each hyperparameter setting model.

Model	Maximum number of iterations	The number of iterations at best RMSE	RMSE	Neurons	Time (s)
ELM-PSO	20	5	1.0387	1462	90000
ELM-BO	400	120	1.0708	1279	129600
ELM-GA	20	6	1.0668	1494	108000

TABLE 3: Prediction errors on each model's test set for the W-G HSR line.

Model	RMSE	MAE	R-squared	Time (s)
ELM-PSO	1.0387	0.3490	0.9955	856
CB	1.6808	1.0464	0.9883	9
GBDT	1.9976	1.2031	0.9835	18
Lasso	1.9852	0.9240	0.9847	9
KNN	1.6488	0.5025	0.9887	27

TABLE 4: Prediction errors on each model's training set for the W-G HSR line.

Model	RMSE	MAE	R-squared
ELM-PSO	0.8247	0.3377	0.9973
CB	1.6368	1.0407	0.9896
GBDT	1.9495	1.2107	0.9852
Lasso	1.6046	0.9199	0.9893
KNN	1.4681	0.4558	0.9916

each stations exceeds 75%. In brief, the overall results confirm the impressive prediction correctness of the proposed model.

At last, we investigate the robustness of our model to data size. In detail, we further train and test our model using 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%, respectively, of the total data as test set, and compare the results with the baseline models. The data sizes used in the experiments are shown in Figure 9. The performance of our model on both training and test set is more outstanding than others. As we can see, the RMSEs of our model stay pretty stable using data with different sizes, while the RMSEs of

other models are higher and fluctuating. These figures show that the proposed model has the smallest predictive RMSE, MAE, and  $R$ -squared for all trains, which demonstrates the robustness of our model to different data sizes.

**5.4. Statistical Tests.** In this section, the Friedman test (FT) and Wilcoxon signed rank test (WSRT) are used to verify the advantages of our proposed method compared with other methods [35, 38]. The results FT and WSRT are shown in Table 6. FT algorithm is a nonparametric statistical tool, which determines the difference by ranking

TABLE 5: Model performance comparison on test set for the five models for different delay bins.

Delay bin (seconds)	Model	RMSE	MAE	R-squared
[0, 1200]	ELM-PSO	0.5201	0.3006	0.9655
	CB	1.2628	0.9223	0.7967
	GBDT	1.3968	0.9886	0.7513
	Lasso	1.0957	0.7903	0.8469
	KNN	1.0195	0.5278	0.8675
>1200	ELM-PSO	5.6009	2.8249	0.9924
	CB	6.2335	3.1986	0.9906
	GBDT	8.0733	4.8589	0.9843
	Lasso	6.5023	3.1169	0.9898
	KNN	9.0247	4.8611	0.9804
All delayed trains (>240)	ELM-PSO	3.3116	1.3457	0.9951
	CB	4.0469	2.2680	0.9927
	GBDT	5.1561	3.1498	0.9881
	Lasso	3.9251	1.7418	0.9931
	KNN	5.5878	2.8828	0.9860

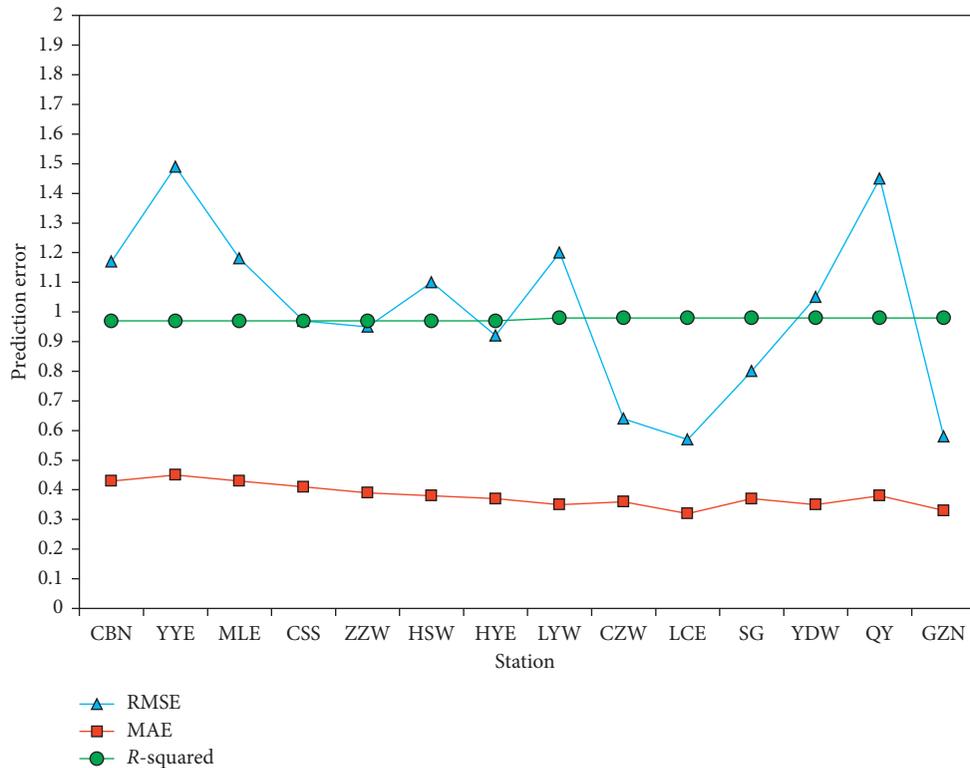


FIGURE 7: Prediction errors in terms of the RMSE, MAE, and R-squared for different stations.

the performance of each method. It can be seen from the table that the proposed method has better ranking than CB, GBDT, Lasso, and KNN at 5% significance level; that is, the efficiency is better. In addition, the results of WSRT showed that the  $p$ -value was less than 0.05 (5%

significance level), which rejected the null hypothesis. It means that there is a statistical difference between the proposed method and other methods. That is, the performance of the proposed method is better than that of other methods.

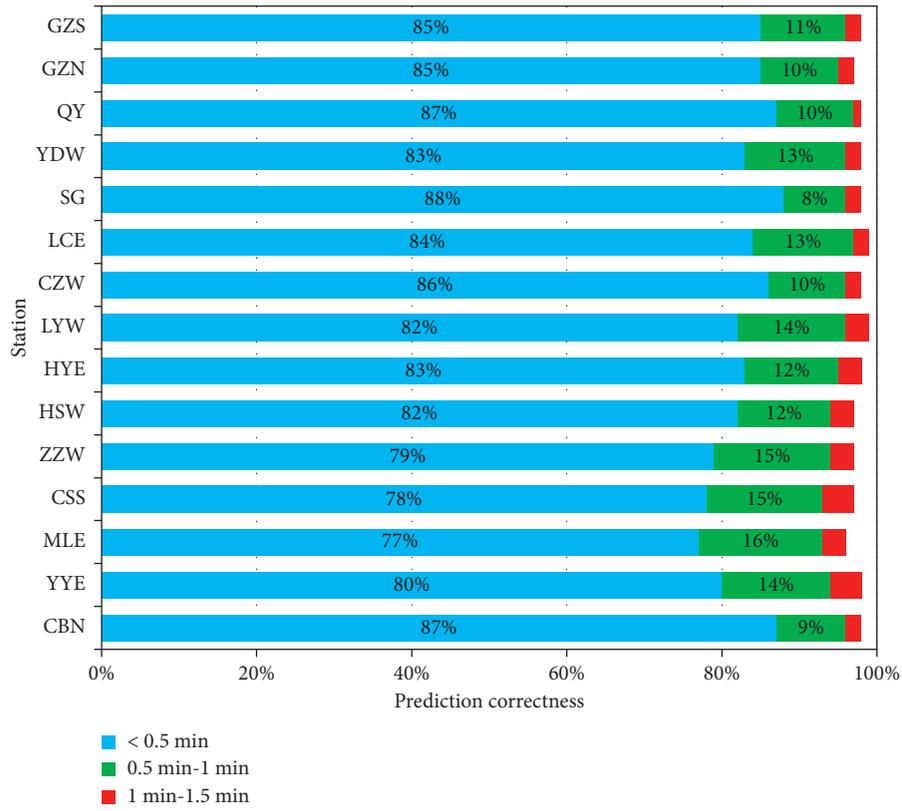
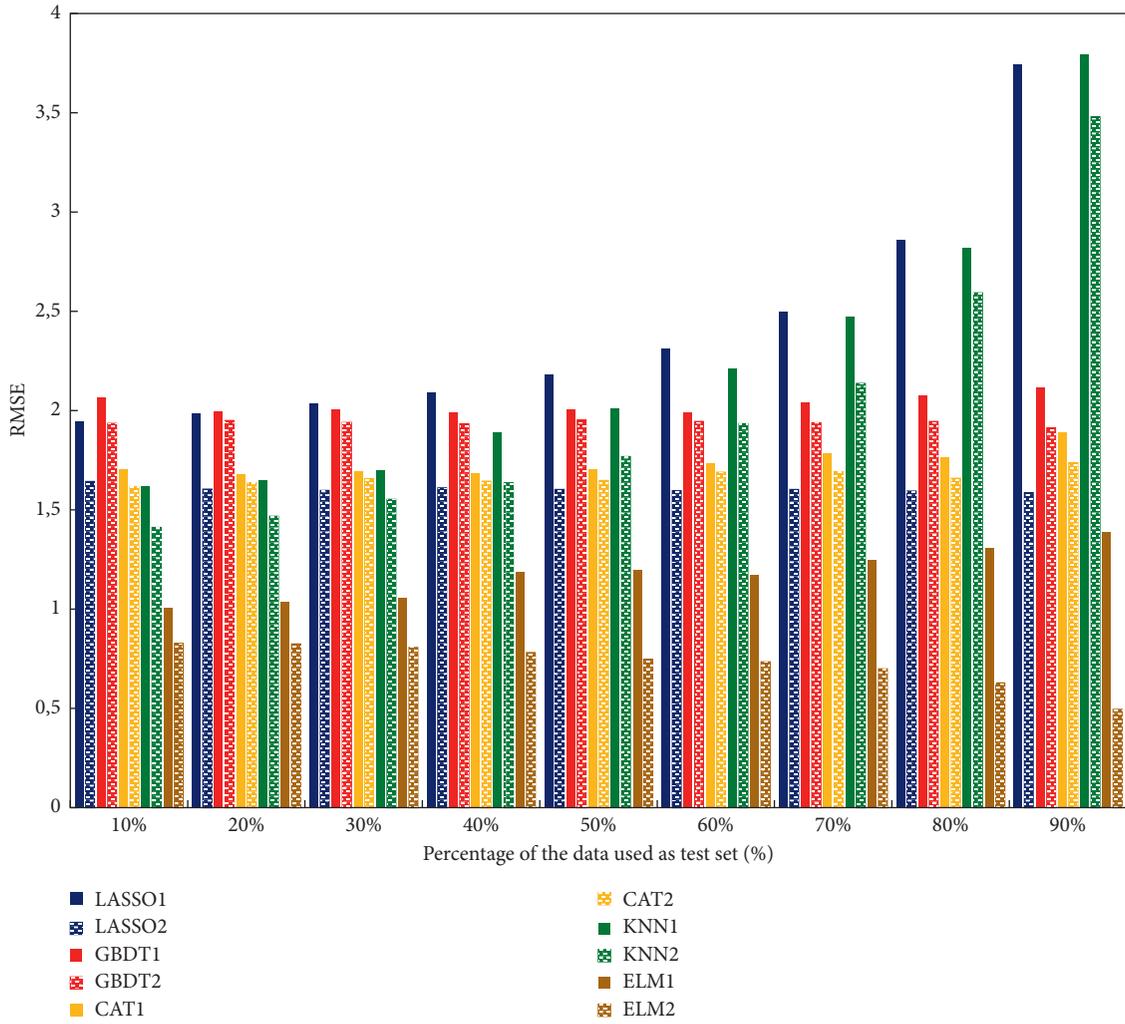
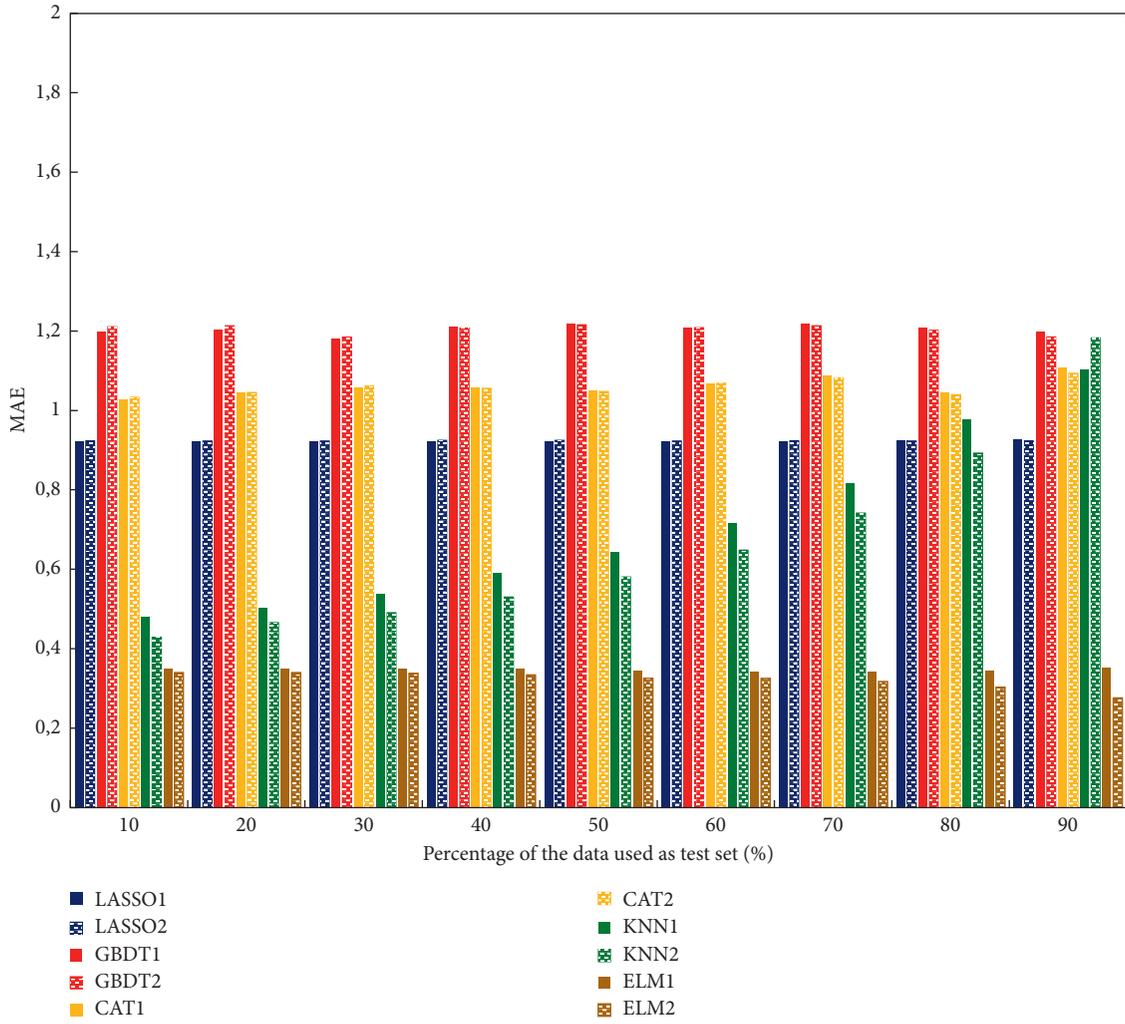


FIGURE 8: Prediction correctness for each station on test set.



(a)

FIGURE 9: Continued.



(b)  
FIGURE 9: Continued.

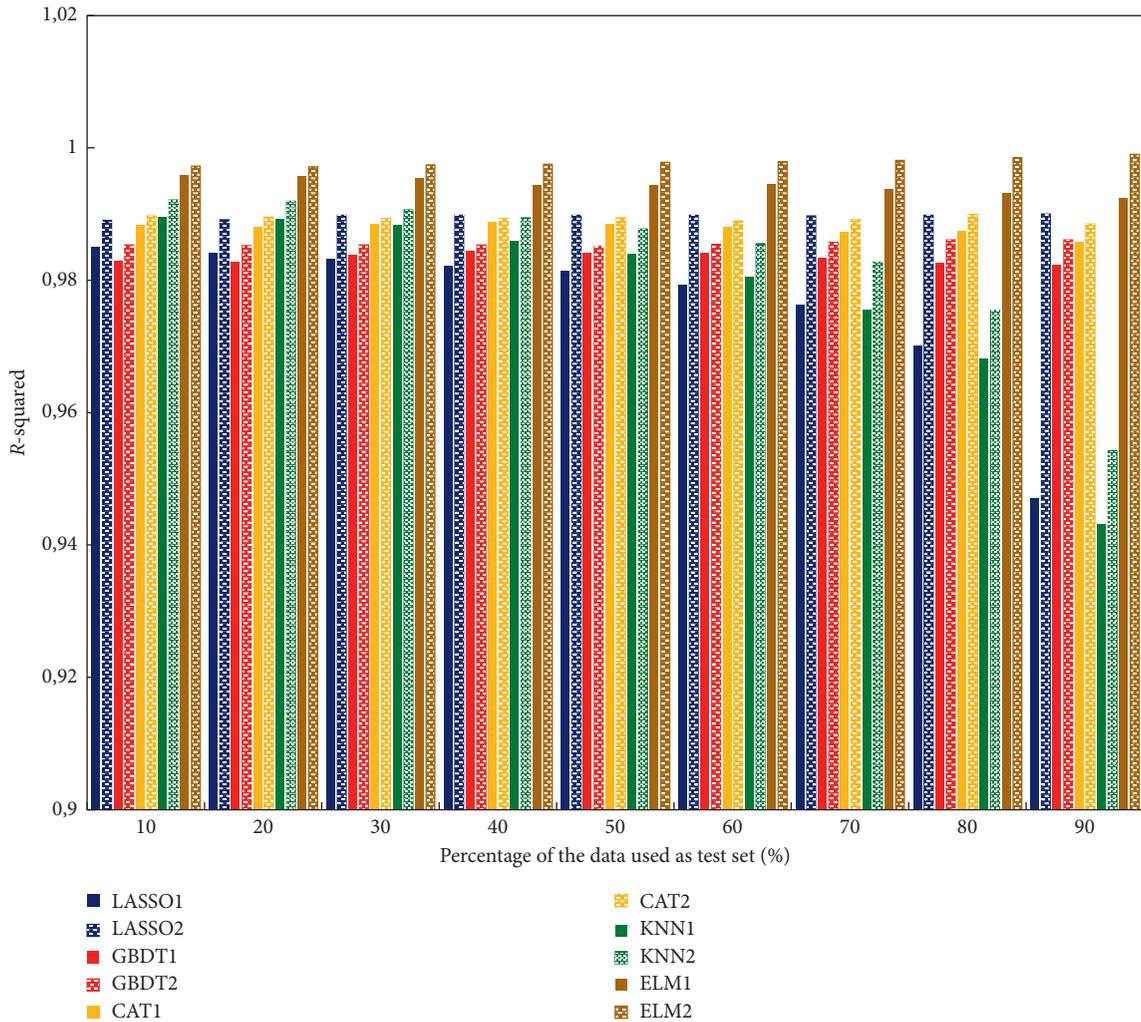


FIGURE 9: MAE, RMSE, and  $R$ -squared values on training and test sets with different data sizes (LASSO1 represents the performance on training set; LASSO2 represents the performance).

TABLE 6: Friedman ranking test and WSRT results.

Models	Mean rank	FT $p$ value	Model 1 vs. model 2	WSRT Z-score	WSRT $p$ value
ELM-PSO (M1)	1.50		—	—	—
CB (M2)	3.55		M1-M2	-259.200	$\leq 0.001$
GBDT (M3)	4.08	$\leq 0.001$	M1-M3	-261.312	$\leq 0.001$
Lasso (M4)	3.53		M1-M4	-257.456	$\leq 0.001$
KNN (M5)	2.34		M1-M5	-174.373	$\leq 0.001$

## 6. Conclusion

In this paper, a hybrid ELM-PSO method is proposed to predict train delays. The ELM can overcome the shortcomings of backpropagation training algorithm, and the advantage of PSO is its excellent ability in searching the best hyperparameter. Four benchmark models, CB, KNN, GBDT, and Lasso models, are selected to compare with proposed model. These models were run on the same data collected from China Railways. ELM-PSO tends to have a

better performance and generalization ability ( $R$ -squared = 0.9955, MAE = 0.3490, RMSE = 1.0387) than the other models on the test set. Our work can not only provide sufficient time and auxiliary decision for the dispatcher to make reasonable optimization and adjustment plan, but also have practical significance for improving the quality of railway service and helping passengers estimate their travel time.

The dataset used in this paper contains train delays under all types of scenarios. Therefore, in the future, we will

consider dividing all the data into certain types of delay scenarios according to particular rules and implementing currently prevalent models to train and predict each scenario to achieve a higher accuracy. Finally, in terms of the input features, all the information of the features in this paper can be obtained from train timetables. In the future, other types of features, such as the infrastructure, weather features, and other HSR lines obstruction, will be taken into account.

## Data Availability

The data used to support the findings of this study were supplied by China Railway Guangzhou Bureau Group Co. Ltd. under license and so cannot be made freely available. Access to these data should be considered by the corresponding author upon request, with permission of China Railway Guangzhou Bureau Group Co. Ltd.

## Conflicts of Interest

The authors declare that they do not have any commercial or associative interest that represents conflicts of interest in connection with the paper they submitted.

## Authors' Contributions

Xu Bao contributed to conceptualization, prepared the original draft, was responsible for software, and visualized the study. Yanqiu Li prepared the original draft, was responsible for software, and visualized the study. Jianmin Li contributed to methodology and reviewed and edited the manuscript. Rui Shi contributed to supervision and data curation. Xin Ding contributed to data curation.

## Acknowledgments

This work was financially supported by the Fundamental Research Funds for the Central Universities of China (2019JBM077) and the Open Fund for Jiangsu Key Laboratory of Traffic and Transportation Security (Huaiyin Institute of Technology).

## References

- [1] J. L. Espinosa-Aranda and R. García-Ródenas, "A demand-based weighted train delay approach for rescheduling railway networks in real time," *Journal of Rail Transport Planning & Management*, vol. 3, no. 1-2, pp. 1-13, 2013.
- [2] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 251-262, 2015.
- [3] C. Wen, W. Mou, P. Huang, and Z. Li, "A predictive model of train delays on a railway line," *Journal of Forecasting*, vol. 39, no. 3, pp. 470-488, 2019.
- [4] A. Higgins and E. Kozan, "Modeling train delays in urban networks," *Transportation Science*, vol. 32, no. 4, pp. 346-357, 1998.
- [5] J. Yuan and I. A. Hansen, "Optimizing capacity utilization of stations by estimating knock-on train delays," *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 202-217, 2007.
- [6] P. B. L. Wiggeraad, *Lighting and Boarding Times of Passengers at Dutch Railway Stations*, TRAIL Research School, Delft, Netherland, 2001.
- [7] M. F. Gorman, "Statistical estimation of railroad congestion delay," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 3, pp. 446-456, 2009.
- [8] P. Huang, C. Wen, and L. Fu, "Modeling train operation as sequences: a study of delay prediction with operation and weather data," *Transportation Research Part E: Logistics and Transportation Review*, vol. 141, 2020.
- [9] P. Kecman and R. M. P. Goverde, "Online data-driven adaptive prediction of train event times," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 465-474, 2015.
- [10] S. Milinković, M. Marković, S. Vesković et al., "A fuzzy Petri net model to estimate train delays," *Simulation Modelling Practice & Theory*, vol. 33, pp. 144-157, 2013.
- [11] J. Peters, B. Emig, M. Jung, and S. Schmidt, "Prediction of delays in public transportation using neural networks," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 2, pp. 92-97, Vienna, Austria, November 2005.
- [12] C. Jiang, P. Huang, J. Lessan, L. Fu, and C. Wen, "Forecasting primary delay recovery of high-speed railway using multiple linear regression, supporting vector machine, artificial neural network, and random forest regression," *Canadian Journal of Civil Engineering*, vol. 46, no. 5, pp. 353-363, 2019.
- [13] J. Hu and B. Noche, "Application of artificial neuron network in analysis of railway delays," *Open Journal of Social Sciences*, vol. 4, no. 11, pp. 59-68, 2016.
- [14] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *Journal of Advanced Transportation*, vol. 47, no. 3, pp. 355-368, 2013.
- [15] F. Corman and P. Kecman, "Stochastic prediction of train delays in real-time using Bayesian networks," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 599-615, 2018.
- [16] J. Lessan, L. Fu, and C. Wen, "A hybrid Bayesian network model for predicting delays in train operations," *Computers & Industrial Engineering*, vol. 127, pp. 1214-1222, 2019.
- [17] P. Huang, C. Wen, L. Fu, Q. Peng, and Y. Tang, "A deep learning approach for multi-attribute data: a study of train delay prediction in railway systems," *Information Sciences*, vol. 516, pp. 234-253, 2020.
- [18] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.
- [19] Y. Chen and W. Wu, "Mapping mineral prospectivity using an extreme learning machine regression," *Ore Geology Reviews*, vol. 80, pp. 200-213, 2017.
- [20] Y. Yoan Miche, A. Sorjamaa, P. Bas et al., "OP-ELM: optimally pruned extreme learning machine," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 158-162, 2010.
- [21] Y. Yimin Yang, Y. Yaonan Wang, and X. Xiaofang Yuan, "Bidirectional extreme learning machine for regression problem and its learning effectiveness," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 9, pp. 1498-1505, 2012.

- [22] F. Han, H.-F. Yao, and Q.-H. Ling, "An improved evolutionary extreme learning machine based on particle swarm optimization," *Neurocomputing*, vol. 116, pp. 87–93, 2013.
- [23] L. Oneto, E. Fumeo, G. Clerico et al., "Dynamic delay predictions for large-scale railway networks: deep and shallow extreme learning machines tuned via thresholdout," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2754–2767, 2017.
- [24] Y. Li and Z. Yang, "Application of EOS-ELM with binary jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092–23101, 2017.
- [25] Y. Zhang, T. Li, G. Na, G. Li, and Y. Li, "Optimized extreme learning machine for power system transient stability prediction using synchrophasors," *Mathematical Problems in Engineering*, vol. 2015, Article ID 529724, 8 pages, 2015.
- [26] Y. Wang, H. Zhang, and G. Zhang, "cPSO-CNN: an efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks," *Swarm and Evolutionary Computation*, vol. 49, pp. 114–123, 2019.
- [27] S. V. Konstantinov, A. I. Diveev, G. I. Balandina, and A. A. Baryshnikov, "Comparative research of random search algorithms and evolutionary algorithms for the optimal control problem of the mobile robot," *Procedia Computer Science*, vol. 150, pp. 462–470, 2019.
- [28] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft Computing*, vol. 22, no. 2, pp. 387–408, 2017.
- [29] Y. Li, X. Xu, J. Li, and R. Shi, "A delay prediction model for high-speed railway: an extreme learning machine tuned via particle swarm optimization," in *Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Rhodes, Greece, September 2020.
- [30] M. Perceptron, J. Tang, and S. Member, "Extreme learning machine for multilayer perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 809–821, 2016.
- [31] J.-R. Zhang, J. Zhang, T.-M. Lok, and M. R. Lyu, "A hybrid particle swarm optimization-back-propagation algorithm for feed forward neural network training," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1026–1037, 2007.
- [32] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.
- [33] Y. Zhang, Z. Zhao, and J. Zheng, "CatBoost: a new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China," *Journal of Hydrology*, vol. 588, Article ID 125087, 2020.
- [34] G. Huang, L. Wu, X. Ma et al., "Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions," *Journal of Hydrology*, vol. 574, pp. 1029–1041, 2019.
- [35] J. Yang, C. Zhao, H. Yu, and H. Chen, "Use GBDT to predict the stock market," *Procedia Computer Science*, vol. 174, pp. 161–171, 2020.
- [36] R. Tibshirani, "Regression shrinkage and selection via the lasso: a retrospective," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, 2011.
- [37] M. Wang, L. Wang, X. Xu, Y. Qin, and L. Qin, "Genetic algorithm-based particle swarm optimization approach to reschedule high-speed railway timetables: a case study in China," *Journal of Advanced Transportation*, vol. 2019, Article ID 6090742, 12 pages, 2019.
- [38] R. Shi, X. Xu, J. Li et al., "Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization," *Applied Soft Computing*, vol. 109, 2021.