A delay prediction model for high-speed railway: an extreme learning machine tuned via particle swarm optimization*

Yanqiu Li, Xinyue Xu, Jianmin Li, and Rui Shi

Abstract— ***Train delay prediction is a significant part of railway delay management, which is key to timetable optimization of High-speed Railways (HSRs). In this paper, an extreme learning machine (ELM) tuned via particle swarm optimization (PSO) is proposed to predict train arrival delays of HSR lines. First, five characteristics (e.g., the plan running time between the present station and the next station, stations) are selected from nine characteristics as input variables for ELM by correlation coefficient matrix. Next, PSO algorithm is implemented to effectively resolve the hyperparameter adjustment of ELM, which overcomes tedious manual regulation for the number of hidden neurons. Finally, a case study of fifteen stations on Beijing-Kowloon (B-K) HSR line in China is proposed using the ELM tuned via PSO (ELM-PSO). The prediction performance of the proposed method is verified by comparison with six benchmark models. The results indicate that our method is superior to these baseline models in prediction accuracy.***

I. Introduction

Ever-advancing changes of high-speed railways (HSRs) is hindered by the train delay problem. On the one hand, it will seriously affect the organization of rail transport and reduce the quality of railway transportation service; on the other hand, it will increase the journey time and gravely affect the prior plan of passengers [1]. Therefore, accurate prediction of HSR delay time can not only enable dispatchers to accurately estimate train operation status and make reasonable decisions [2], but also help passengers estimate travel time [3].

There are numerous delay prediction models established to predict train delay [1]. Traditional method is a kind of classical method such as, analytic models [4], [5] and regression models [6], [7]. Analytic models mainly refer to obtaining the delay probability distribution depending on the targeted data of train delay such as, Weibull, chi-squared, Erlang, exponential distributions [4], and log-normal [5]. Multiple regression models are utilized for train delays predicting with analyzing the linear relationship between multiple independent variables (i.e., influencing factors) and dependent variables (i.e., delay types) [1]. For examples, Wiggenraad et al. [6] chose seven correlative factors (e.g., type of station, train service, and other) to predict the train dwell time. Gorman et al. [7] identified three kinds of factors (i.e., free running time predictors, congestion-related factors and primary congestion predictive factors) to predict train running time, applying linear regression. However, the insurmountable defect of analytical model and regression model is that they require to be updated continually according to train data [8].

Fortunately, with the tremendous upsurge of machine learning methods, train delay prediction methods are becoming more and more abundant, which remedy the defects of traditional methods [1]. Among them, some researchers are devoted to artificial neural networks (ANN) to predict train delays [9]–[12]. However, the prediction accuracy of ANN is not sufficient enough, along with the extraneous and cumbersome parameter adjusting process, to cope with train delay prediction problems. The ELM algorithm based on Moore–Penrose generalized inverse matrix theory was proposed by Huang et al. [13] to address this problem. The algorithm is able to generate the weight and deviation of the input layer randomly, and the output weight of the ELM can be reckoned analytically in only one step. Further, Oneto et al. [14] predicted train delays by both shallow and deep ELMs tuned via the thresholdout technique. We need to explore more contemporary and promising tools based on big data that can effectively adjust hyperparameter.

Therefore, this paper proposed ELM-PSO to predict train arrival delays of HSR. The tedious and time-consuming problems of manually adjusting parameters and low tuning efficiency are solved. The contributions this paper makes are as follow: 1) ELM-PSO is proposed to predict train arrival delays of HSR lines for the first time. 2) The proposed method has higher prediction accuracy than many existing methods. 3) Tested available on large scale data instances in the B-K HSR line of China.

This paper is organized as follows: In Section II, the principle of ELM and PSO is introduced in detail. The train delay data and selection of characteristic variables are described in Section III. In Section IV, the experimental

*Resrach supported by by the national key research project Railroad Comprehensive Efficiency and Service Level Improvement Technology with High Speed Railway Network (Grant No.2018YFB201403) in China *(Corresponding author: Xinyue Xu.)*.

Yanqiu Li, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (phone:+86 15172456305; e-mail: 19125741@bjtu.edu.cn) .

Xinyue Xu, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (corresponding author to provide phone: (010) – 51688547; email: xxy@bjtu.edu.cn).

Jianmin Li, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (phone:+86 15172456305; e-mail: 19114041 @ bjtu.edu.cn) .

Rui Shi, is with State Key Laboratory of Railway Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (phone:+86 19801211681; e-mail: 564619110@qq.com) .

settings and performance of the proposed method results are discussed. The conclusions are presented in the final section .

II. Methods

A. Extreme Learning Machine

ELM was initially proposed by Huang et al. [13] as a single hidden layer feedforward neural network, which aims to overcome the problems caused by the back-propagation training algorithm: the potential slow convergence speed, the critical adjustment of optimization parameters, and the existence of local minima that need to be retrained.

For this kind of neural network, let us suppose that P represents the hidden layer node and x_j with $j \in \{1, \ldots, n\}$ implies the input of the input layer.

$$f_j(x) = \sum_{i=1}^{P} \beta_{i,j} g\left(\sum_{j=1}^{n} w_{i,j} x_j + b_i\right), \quad i=1, \ldots, P \tag{1}$$

where $g(x)$ is the activation function, $\beta_{i,j}$ is the value of output weight, $w_{i,j}$ is the weight of the ith hidden layer neuron and the jth input, and b_i is the offset of the ith hidden layer.

Replace the above function with the value to be predicted, the equation can be expressed as:

$$y_j = \sum_{i=1}^{P} \beta_{i,j} g\left(\sum_{j=1}^{n} w_{i,j} x_j + b_i\right) \tag{2}$$

The above Eq. (2) can be expressed as a matrix:

$$y_j = \sum_{i=1}^{P} \beta_{i,j} g_i(x), \quad i=1, \ldots, P \tag{3}$$

$$D = \left[g_1(x) \cdots g_P(x)\right] \tag{4}$$

Owing to the random setting of the weight $w_{i,j}$ and the deviation b_i, the quantity $\beta_{i,j}$ in Eq. (2) is the only unknown quantity.

In this case, the problem is transformed into the minimization of the convex cost:

$$\beta^* = argmin \, || D\beta\text{-}Y ||^2 \tag{5}$$

where $Y = \left[y_1, y_2, \ldots, y_j\right]$, $\beta = \begin{bmatrix} \beta_{1,1} \cdots \beta_{1,n} \\ \vdots \quad \cdots \quad \vdots \\ \beta_{P,1} \cdots \beta_{P,n} \end{bmatrix}$.

Matrix pseudoinversion yields the unique solution:

$$\beta^* = D^+ Y \tag{6}$$

where D^+ is the Moore–Penrose matrix pseudoinverse of matrix D.

B. Particle swarm optimization

The advantage that PSO possesses is its understandability. At present, it has been proverbially employed to optimize the neural network parameter, and can perform global searches in the solution space [15]. In PSO, the potential resolution of every parameter optimization problem represents a bird in the searching space, a bird refers to as a particle. All particles have a fitness function to judge the performance of each particle, and each particle is defined by its position and velocity. Then, the particles follow the current optimal particle to keep locating and determining the best particle [16].

Each possible solution in the PSO algorithm is called a particle (x_i), whereas each particle also has a measure of quality of each particle ($fitness$). The fitness values of particles influences the global search mechanism of the PSO through the calculation of the velocity (v_i), which is shown in Eq. (13). The algorithm keeps records of the best particle in the swarm ($gbest$), and each particle records its personal best solution ($pbest$).

Let us suppose that in a D -dimensional target searching space, there is a community of N particles. The position of the ith particle is a vector that includes D dimensions, which is recorded as:

$$X_i(ite) = \left[x_{i_1}(ite), \ldots, x_{i_D}(ite)\right], \; i = 1, \ldots, N \tag{7}$$

The same goes to the velocity:

$$V_i(ite) = \left[v_{i_1}(ite), \ldots, v_{i_D}(ite)\right], i = 1, \ldots, N \tag{8}$$

where i denotes the number of particles. ite represents the number of iterations of the PSO algorithm. At each iteration, there are D positions of each particle. The fitness function is expressed as $p_{i_k}(ite), k = 1, \ldots, D$.

The optimal fitness of the ith particle searched so far is called $pbest$, which is recorded as:

$$pbest(ite) = optimal\left[p_{i_1}(ite), \ldots, p_{i_D}(ite)\right] \tag{9}$$

The optimal fitness of the whole particle swarm so far is referred to as $gbest$.

$$gbest(ite) = optimal\left[p_{j_1}(ite), \ldots, p_{j_D}(ite)\right] \tag{10}$$

PSO can find $gbest$ in the first iteration, and then compare it with the obtained $gbest$ in the next iteration. The optimal $gbest$ is defined as the best $fitness$. Therefore, if $p_i(ite) < p_i(t), \; (t = 0, 1, \ldots, ite)$, then $p_{i_k pbest} = p_i(ite)$. Additionally, if $p_i(ite) < p_{i_k pbest}(ite)$, then the global best position will be defined as $p_{gbest} = p_{i_k pbest}$. The PSO algorithm keeps adjusting the parameter until reaching the maximum number of iterations.

As a result, the position of each particle will be updated as below.

$$v_{i_k}\left(ite\right) = wv_{i_k}\left(ite-1\right) + c_1 g_1\left(p_{i_k pbest}\left(ite-1\right) - p_{i_k}\left(ite-1\right)\right) + c_2 g_2\left(p_{gbest}\left(ite-1\right) - p_{i_k}\left(ite-1\right)\right) \tag{11}$$

$$p_{i_k}\left(ite\right) = v_{i_k}\left(ite\right) + p_{i_k}\left(ite-1\right) \tag{12}$$

where c_1 and c_2 are learning factors, w is the inertia factor, g_1 and g_2 are two random numbers, whose range are both between 0-1.

C. *Combination model*

The proposed combination model is as follows. A particle is a possible solution to the combination system. Each hyperparameter of the ELM equals to a particle of PSO. Only one hyperparameter needs to be adjusted in the ELM: the number of neurons in the hidden layer. Therefore, the number of hidden neurons of ELM that needs to be adjusted is regarded as the particle of PSO, and the search dimension is merely 1. Multiple particles are combined into a population of PSO. The fitness of each particle in the population is calculated by Eq. (13) predicted by ELM on test set, and the lowest fitness is the best fitness of this population. Then it keeps searching the next population and calculate the best fitness of the next population at the same time. PSO will update the best fitness compared with the best fitness of the previous population. The PSO algorithm keeps searching until reaching the maximum number of iterations, eliminating the tedious debugging for the hidden neurons. The fitness function is calculated according to Eq. (13), which equals to root mean squared error (RMSE).

$$fitness = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{13}$$

III. Experimental Study

A. *Dataset description*

The data applied to this study is extracted from the train operation records on B-K HSR line in China. The part of the B-K HSR line comprises fifteen stations, as depicted in Fig. 1. More than 400000 train operation records are obtained from October 2018 to April 2019. The operation data records the train number of each train every day, arrival (departure) and passing station, nominal timetable and actual timetable of each train, maximum train operation speed, and so on.

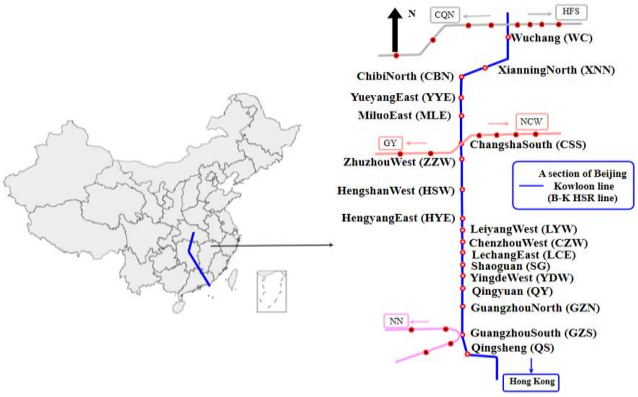

Figure 1. The overview of a section of the B- HSR line

The arrival delay of main stations from GZS to CBN is statistically analyzed, such as SG, HYE, and ZZW. The analysis results are shown in Table Ⅰ. The following conclusions can be drawn: 1) SG and ZZW account for the highest proportion of train delay, and they are the stations with the most serious train delay; 2) the number of trains delayed in HYE is the most, and the total delay time reaches 168388 minutes in half year. In a word, the station is the main place of train delay and the bottleneck of network operation.

TABLE I. STATISTICS OF TRAIN ARRIVAL DELAY

Station	PD (>4min)	PD (>10min)	NDT	TDT (min)
GZS	0.04	0.025	2589	87933
SG	0.082	0.038	2918	76076
HYE	0.07	0.036	6029	168388
ZZW	0.082	0.038	3221	83627
YYE	0.045	0.030	1441	62514
CBN	0.041	0.029	1237	58232

PD: Proportion of delay, NDT: Number of delay trains, TDT: Total delay time

B. *Selection of characteristic variables*

Based on data analysis of existing data, nine factors that may affect the train arrival delays are selected. The nine independent input variables are as follows:

1. The actual running time between the present station and the next station (X_1).
2. The distance between the present station and the previous station (X_2).
3. The plan running time between the present station and the previous station (X_3).
4. The train number (X_4).
5. The plan running time between the present station and the next station (X_5).
6. The actual running time between the present station and the previous station (X_6).
7. Stations (X_7).
8. The distance between the present station and the next station (X_8)
9. The arrival delay time at present station (X_9).

Y represents the arrival delay time at the next station.

The variables selected in this paper are the common influencing factors of train delay prediction problem. The importance of the nine variables is arranged by correlation coefficient matrix to filter extra variables, and the correlation between train delays and the nine input variables is exhibited below in Fig. 2. X_9 and X_7 have statistically significant correlations with Y. Finally, we select X_9 , X_7, X_6, X_5 and X_4 which are in the top five in importance as the final variables to avoid over-fitting.

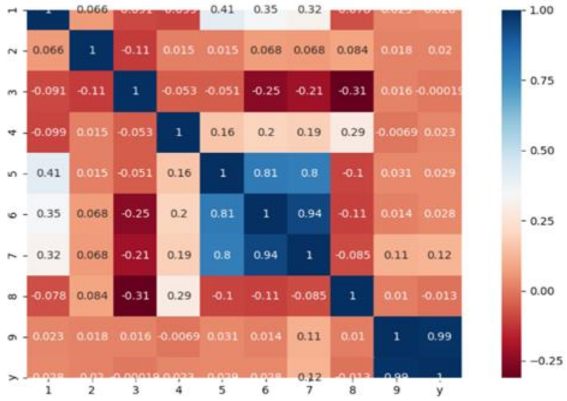

Figure 2. Heat map of the correlation between the input variables and output

IV. Results

A. Model settings

To evaluate the prediction accuracy of the models, three prevalent error indices are used. Supposing that $y_1, y_2, \cdots, y_n$ are actual values, $\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n$ are predicted values, and $\overline{y}$ is the mean value of y_i , the calculation process of the three indices is as follows:

$$R\text{-}squared = 1 - \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N} (y_i - \overline{y})} \tag{11}$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{y}_i - \overline{y} \right| \tag{12}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{13}$$

Six classic models were chosen as baseline models to assess the feasibility of our model. These models ran exactly the same as our model to eliminate errors. Briefly, an overview description of each model is as follows:

(1) KNN (k-nearest neighbor): As a burgeoning and highly flexible model, the KNN model is employed as a baseline model.

(2) ANN(artificial neural networks): ANNs are outstanding modelling technologies, especially for datasets with nonlinear relationships. Thus, an ANN is also employed as a benchmark.

(3) Decision Tree: Decision trees are usually easy to understand and master and are regarded as a baseline model.

(4) Lasso: Lasso is capable of simultaneously performing regularization and variable filtering, which is crucial to the benefit of Lasso.

(5) GBDT (gradient boosting decision tree): GBDT generates a weak classifier through multiple iterations, each of which is trained on the basis of the residual of the previous one.

(6) XGB (extreme gradient boosting): XGB is one of the boosting algorithms. The idea of boosting algorithm is to integrate many weak classifiers to form a strong classifier.

To find the best hyperparameter of ELM, the parameters settings of PSO are as follows:

- *Maximum number of iterations = 10*
- Particle number = 40
- *Search dimension = 1*
- *Particle search range = [1,1000]*
- *Fitness function: RMSE on test set*

B. Performance results

We directly use the train-test split method provided by sklearn to divide the data set into test set and training set according to the proportion of 2:8. The experiment is implemented in an environment with a processor Intel® Core i5-6200U, 2.13 GHz, memory size 8 G via Python.

The process of the hyperparameter becoming optimal is to make the fitness value of test set minimum, displayed in Fig. 3. The results indicate that the fitness achieves minimum after six iterations. The best fitness value is 2.13 when there are 996 neurons of the ELM and the structure of the network is optimal at this time. And in such a large amount of data, the total time spent is only a few hours.

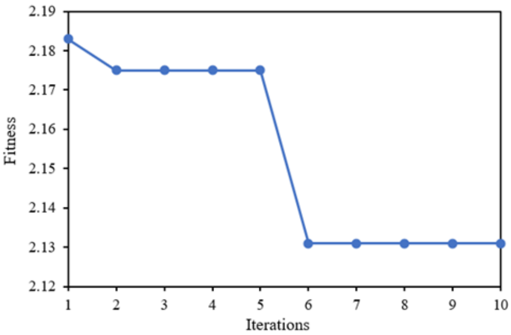

Figure 3. Convergence discriminant graph of PSO optimization

The corresponding results of test set are summarized in Table II, which presents the prediction errors of each model for the B-K HSR line. We can see that our model outperforms the other six comparative models in all three metrics (R-squared=0.98, MAE=0.89, RMSE=2.13). Compared with the traditional model ANN, the RMSE of ELM-PSO is 2.13, almost half of that of ANN. Compared with the contemporary popular method XGB, our model performs better in both MAE and RMSE. For the other four models, including machine learning (KNN and DT) and regularized regression model (Lasso), although the performance advantage of R-squared is not great. The R-squared of ELM-PSO is 0.98, and their R-squared values are all 0.97. But there is a big gap in RMSE performance. It is proved that this model is still outstanding in comparison with many different methods. We can draw a conclusion that ELM-PSO outperforms other methods under the three indexes comprehensively, and is able to be effectively applied to the actual railway operations.

TABLE II. PREDICTION ERRORS OF EACH MODEL'S TEST SET FOR THE B-K HSR LINE (MINUTE).

Model	RMSE	MAE	R-squared
ELM-PSO	2.13	0.89	0.98
KNN	3.97	0.86	0.95
ANN	4.77	1.12	0.92
Decision Tree	3.06	1.48	0.97
Lasso	2.60	1.40	0.97
GBDT	3.20	1.52	0.96
XGB	2.14	0.92	0.98

V. CONCLUSION

This paper presented the first application of ELM-PSO in the train delay prediction problem. ELM is able to overcome the problems caused by the back-propagation training algorithm. The preponderance of PSO is understandability and high efficiency. The combination of the two can accurately and efficiently predict the train arrival delay in the case of big data. Six benchmark models: ANN, KNN, Decision Tree, Lasso, GBDT and XGB, were chosen to compare with our model. ELM-PSO turned out to have better performance (MAE = 0.89, RMSE = 2.13 and R-squared = 0.98) than other models. Moreover, our method has good performance in accuracy. It can be used to guide the dispatchers to arrange the train transportation organization in case of delay problem. This result is conducive to further applications of the ELM in train delay predictions.

REFERENCES

[1] N. Marković, S. Milinković, K. S. Tikhonov, and P. Schonfeld, "Analyzing passenger train arrival delays with support vector regression," *Transp. Res. Part C Emerg. Technol.*, vol. 56, pp. 251–262, 2015.

[2] C. Wen, W. Mou, P. Huang, and Z. Li, "A predictive model of train delays on a railway line," *J. Forecast.*, pp. 0–2, 2019.

[3] P. Wang and Q. Zhang, "Train delay analysis and prediction based on big data fusion," *Transp. Saf. Environ.*, vol. 1, no. 1, pp. 79–88, 2019.

[4] R. W. Hall, "Transportation Science," no. January 2015, pp. 1–4, 1999.

[5] J. Yuan and I. A. Hansen, "Optimizing capacity utilization of stations by estimating knock-on train delays," *Transp. Res. Part B Methodol.*, vol. 41, no. 2, pp. 202–217, 2007.

[6] P. B. L. Wiggenraad, "Alighting and boarding times of passengers at Dutch railway stations," 2001.

[7] M. F. Gorman, "Statistical estimation of railroad congestion delay," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 45, no. 3, pp. 446–456, 2009.

[8] P. Kecman and R. M. P. Goverde, "Online data-driven adaptive prediction of train event times," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 465–474, 2015.

[9] J. Peters, B. Emig, M. Jung, and S. Schmidt, "Prediction of delays in public transportation using neural networks," *Proc. - Int. Conf. Comput. Intell. Model. Control Autom. CIMCA 2005 Int. Conf. Intell. Agents, Web Technol. Internet*, vol. 2, pp. 92–97, 2005.

[10] S. Pongnumkul, T. Pechprasarn, N. Kunaseth, and K. Chaipah, "Improving arrival time prediction of Thailand's passenger trains using historical travel times," in *2014 11th Int. Joint Conf. on Computer Science and Software Engineering, JCSSE 2014*, 2014.

[11] H. Li, Y. Wang, X. Xu, L. Qin, and H. Zhang, "Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network," *Appl. Soft Comput. J.*, vol. 83, p. 105620, 2019.

[12] M. M. K. and M. S. Masoud Yaghini*, "Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data," *J. Adv. Transp.*, vol. 47, no. June 2010, pp. 512–525, 2013.

[13] 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.

[14] L. Oneto *et al.*, "Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 10, pp. 2754–2767, 2017.

[15] J. F. L. de Oliveira, L. D. S. Pacífico, P. S. G. de Mattos Neto, E. F. S. Barreiros, C. M. de O. Rodrigues, and A. T. de A. Filho, "A hybrid optimized error correction system for time series forecasting," *Appl. Soft Comput. J.*, vol. 87, p. 105970, 2020.

[16] 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," vol. 2019, pp. 13–16, 2019.