Received September 10, 2019, accepted September 27, 2019, date of publication September 30, 2019,
date of current version October 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944744

Prediction of Passenger Flow in Urban Rail Transit Based on Big Data Analysis and Deep Learning

KAER ZHU, PING XUN, WEI LI, ZHEN LI, AND RUOCHONG ZHOU
Department of Military Facilities, Army Logistics University of PLA, Chongqing 401311, China
Corresponding author: Kaer Zhu (zhu_kaer@163.com)

ABSTRACT Passenger flow prediction is the key to operation efficiency and safety of urban rail transit (URT). This paper combines the deep learning (DL) theory and support vector machine (SVM) into the DL-SVM model for URT passenger flow prediction. Firstly, the deep belief network (DBN) was adopted to extract the features and inherent variation of passenger flow data. On this basis, an SVM regression model was constructed to predict passenger flow. Then, the proposed model was compared with three shallow prediction models through experiments on Qingdao Metro. The results show that the DL-SVM outperformed the other models in accuracy and stability. The research findings shed important new light on the passenger flow prediction in the URT system.

INDEX TERMS Big data analysis, deep belief network (DBN), support vector machine (SVM), passenger flow prediction, urban rail transit (URT).

I. INTRODUCTION

Urban rail transit (URT) is a fast, safe and comfortable mode of transport in urban traffic network. The URT efficiency depends heavily on parameters like passenger volume, vacancy rate and speed. With the continuous accumulation of these parametric data, it is now possible to predict the passenger flow of the URT through big data analysis.

The passenger flow of the URT [1] refers to the number of people entering metro stations and related facilities per unit time. This parameter is critical to the rational planning of metro traffic network, allocation of metro stations and related facilities, and formulation of metro operation plan. The URT system may not satisfy the travel demand of passengers, if the passenger flow is not predicted accurately.

The passenger flow within a short period is known as the short-term passenger flow [2]. Within the short period of time, the passenger flow appears and disappears rather quickly and randomly, making it hard to grasp the law of its occurrence. Compared with long-term travel demand, the short-term travel demand of passengers is difficult to fulfill through the URT operations. After all, the train frequency, train speed and line condition vary from time to time. This calls for accurate prediction of short-term passenger flow in the URT, that is, the calculation of the passenger flow in the next or more periods based on the variation law and intrinsic relationship of the historical data.

Currently, the short-term passenger flow of the URT is often examined within the periods of 5min, 10min or 15min. The shorter the period, the more random the passenger flow per unit time, and the harder it is to predict the short-term passenger flow. In this paper, the deep learning (DL) theory and support vector machine (SVM) are integrated into the DL-SVM prediction model, and applied to predict the exact short-term passenger flow of the URT.

II. LITERATURE REVIEW

Short-term passenger flow is often predicted by linear models. In these models, the prediction function is obtained through linear mapping of historical data, aiming to approximate the trend of historical data. For example, Kumar and Vanajakshi [3] constructed an autoregressive integrated moving average (ARIMA) model to forecast short-term traffic flow, which estimates the parameters using the recursive least squares method with forgetting factor. Ye *et al.* [4] established an ARIMA model with 7-day average passenger flow, and relied on it to project the daily passenger flow in Shanghai Metro. Haworth *et al.* [5] analyzed the stationary

The associate editor coordinating the review of this manuscript and approving it for publication was Sabah Mohammed.

and periodic features of autocorrelation function and partial autocorrelation function, eliminated trend and periodicity from the original data, and put forward an ARIMA model to predict metro passenger flow based on the product between periodicity and short-term correlation of the processed data. Wang *et al.* [6] forecasted short-term passenger flow at metro stations with both backpropagation neural network (BPNN) and Kalman filter, revealing that the Kalman filter achieved relatively small mean absolute error (MAE) and mean squared error (MSE). Zhang *et al.* [2] developed an adaptive Kalman filter based on the original Kalman filter, the seasonal-difference autoregressive moving average (ARMA) model and the generalized autoregressive conditional heteroscedasticity (ARCH) model, and experimentally verified that the adaptive Kalman filter is a desirable tool to solve short-term passenger flow problems with high fluctuation.

In real-world URT systems, passenger flow varies greatly with the elapse of time, and the relevant data are highly nonlinear and stochastic. The short-term variation of metro passenger flow can be illustrated well with nonlinear models based on wavelet theory, chaotic theory and nonparametric regression method. For instance, Sun *et al.* [7] combined wavelet theory and neural network into a wavelet neural network that can forecast passenger flow and traffic flow in an accurate manner. Considering the self-similarity, singularity and fractality of traffic flow data, Lin *et al.* [8] constructed a non-parametric dynamic time-delay recursive wavelet neural network, and experimentally proved the long-term and short-term prediction effects of the network. Jiang *et al.* [9] decomposed the intraday passenger flow data into sequences of multiple scales, observed the chaotic features of the data, and reconstructed the phase space. Li *et al.* [10] used the Lyapunov exponent to prove the obvious chaotic features of passenger/traffic flow data, and then processed passenger flow data according to phase space reconstruction theory and fractal theory. Hou *et al.* [11] estimated the short-term traffic flow by the k-nearest neighbors (k-NN) model and the linear time series model, and learned that the k-NN model has certain advantages over the other model.

More and more machine learning methods have been adopted to handle the various influencing factors of metro passenger flow. For example, Wei and Chen [12] introduced the theory of empirical mode decomposition into BPNN, and then divided to prediction of short-term passenger flow in the URT into three phases. Chen and Wei [13] compared the performance of neural network and the ARIMA model in passenger flow prediction. Yang and Hou [14] proposed a recurrent neural network with delays, which can forecast passenger flow in the next 5min with a greater-than-90 % accuracy.

Along with the stricter demand of prediction accuracy, shallow learning (SL) machines have been gradually replaced with DL machines in relevant studies. The original data are often subjected to multi-level feature extraction, making it easier for the learning machines to decipher the inherent law. For instance, Lv *et al.* [15] designed a DL prediction model in the light of the inherent time-space relationship in traffic flow data. Huang *et al.* [16] acquired the features of traffic flow data through unsupervised learning, using belief network model and multitask regression model.

III. MODEL CONSTRUCTION

In this paper, we design and implement DL-SVM prediction model based on deep learning theory and support vector regression theory. The DL-SVM model consists of a DBN model in the lower part and an SVM regression model in the upper part. The DBN model extracts the features from the historical data on short-term passenger flow of the URT, while the SVM regression model predicts the short-term passenger flow of the URT. The robustness and stability of the DL-SVM model was enhanced by unsupervised weighting and supervised finetuning.

A. THE SVM REGRESSION MODEL

Let $T = \{(x_i, y_i), i = 1, 2, \ldots, \mathrm{Z}\}, x_i \in R, y_i \in R$ be the set of Z training samples. The SVM-based regression takes place in three steps: setting up the nonlinear mapping relationship between input and output, mapping the original data to a high-dimensional feature space, and processing the feature space with regression equation $f(x)$.

If the data are linearly inseparable, nonnegative relaxation variables (δ and δ^*), penalty factor (P) and insensitive loss (ε) will be introduced to convert the data into linearly separable form. Then, the optimization problem can be defined as:

$$\begin{aligned} &\min \frac{1}{2}\|\omega\|^2 + \mathrm{P}\sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}}(\delta_{\mathrm{i}} + \delta_{\mathrm{i}}^*) \\ &\text{s.t. } y_i - (\omega x + b) \le \varepsilon + \delta_i, \quad (\omega x + b) - y_i \le \varepsilon + \delta_i^*, \\ &\quad \delta, \quad \delta^* \ge 0, \ i = 1, 2, \ldots, n \end{aligned} \tag{1}$$

The optimization problem can also be written as:

$$\begin{aligned} &\max \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{a_i} - \frac{1}{2}\sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}}\sum\nolimits_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{a_i a_j x_i x_j y_i y_j} \\ &\text{s.t. } \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{a_i y_i} = 0, \quad 0 \le \mathrm{a_i} \le \mathrm{P(i = 1, 2, \ldots, n)} \end{aligned} \tag{2}$$

Kernel function needs to be integrated into the SVM to prevent the curse of dimensionality in high-dimensional space. The common kernel functions are listed in Table 1 below.

The performance of the SVM regression model directly hinges on the selection of the penalty factor, the kernel function and the insensitive loss. In this paper, the combination of parameters is optimized by a heuristic algorithm.

In order to obtain the optimal combination of parameters, genetic algorithm is used as the parameter optimization method to optimize the parameters of support vector regression machine. Specifically, the parameter combination (P, σ) was optimized by the genetic algorithm (GA) [17]–[20]. The GA, known for its powerful search ability, mainly includes a coding plan, a fitness function and several genetic operators.

TABLE 1. Common kernel functions.

Kernel functions	Expressions
Linear kernel	$K(x_i, x_j) = x_i \cdot x_j + P$
Polynomial kernel	$K(x_i, x_j) = (x_i \cdot x_j + P)^d, d \geq 1$
Sigmoid kernel	$K(x_i, x_j) = \tan^{-1}(\beta(x_i \cdot x_j) + P), \beta > 0$
Gaussian kernel	$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \ , \sigma > 0$
Laplace kernel	$K(x_i, x_j) = \exp\ (-\frac{\|x_i - x_j\|}{\sigma}), \sigma > 0$

1) CODING

Firstly, the parameter combination is expressed as a set of chromosomes in the form of random binary numbers. Each binary number represents a genetic locus. The binary coding lengths of P and σ can be respectively computed by:

$$P_{length} = \log_2[(P_{max} - P_{min}) * 100] \tag{3}$$

$$\sigma_{length} = \log_2[(\sigma_{max} - \sigma_{min}) * 100] \tag{4}$$

2) FITNESS CALCULATION

The fitness function ensures that GA can iteratively search for the optimal solution. During the iterative optimization, the fitness of each parameter combination was computed by the fitness function, and the one of the highest fitness is taken as the optimal parameter combination of the current generation. When the fitness remains unchanged or the maximum number of iterations has been reached, the GA is considered to have converged to the optimal solution. In this paper, the MSE is adopted to evaluate the quality of the optimal solution of the GA:

$$MSE = \frac{1}{T}\sum_{i=1}^{Z}(f(x_i) - y_i^*)^2$$

where, T is the number of training samples; $f(x_i)$ is the predicted value; y_i^* is the actual value.

The fitness function can be defined as:

$$Fit\,(P, \sigma) = \begin{cases} \frac{1}{MSE}, & MSE \neq 0 \\ Fit_{Best} + 1, & MSE = 0 \end{cases}$$

where, Fit_{Best} is the fitness of the optimal parameter combination (P, σ).

If the MSE of a set of solutions is very small, then the SVM regression model trained by the parameters can make close-to-reality predictions, that is, the current parameter combination (P, σ) has high adaptability.

3) GENETIC OPERATIONS

The GA continues to converge to the optimal solution through three genetic operations, namely, selection, crossover and mutation. Meanwhile, the population diversity is maintained to prevent the algorithm from falling into the local optimum trap.

The selection mainly retains the high-quality parameter combination (P, σ), and takes it as the parent of the next generation, such as to continuously improve the overall quality of the solution population.

In this paper, roulette selection is employed to select the parent (P, σ) parameter combination. The probability P_i of a parameter combination to be selected as the parent was measured by the training fitness:

$$P_i = \frac{Fit_i}{\sum_i^N Fit_i}$$

$$Q_i = \sum_{k=1}^{i} P_k, \quad k = 1, 2, \ldots i$$

where, N is the population size; Fit_i is the fitness of parameter combination i; Q_i is the cumulative probability of parameter combination i. A random threshold $R \in (0, 1)$ was introduced as the judgement criterion. If $Q_{i-1} < R < Q_i$, then parameter combination i will be selected as the parent for the new generation.

4) ALGORITHM FLOW

The GA-based optimization of parameter combination (P, σ) can be summed up as:

Step 1: Initialize the parameters of the GA and the SVM regression model, such as the crossover probability, the mutation probability, the maximum number of iterations, the insensitive loss and population size.

Step 2: Determine the range of the parameter combination (P, σ).

Step 3: Establish the $N \times (P_{length} + \sigma_{length})$ matrix and randomly initialize the population.

Step 4: Calculate the fitness $Fit(P, \sigma)$ of each parameter combination.

Step 5: Determine the parameter combinations that are eligible for selection.

Step 6: Perform crossover and mutation of the selected parameter combinations and update the population according to the results.

Step 7: Judge if the GA has converged the optimal solution or if the maximum number of iterations has been achieved. If yes, output the optimal solution; Otherwise, repeat Steps 4~7.

Step 8: Use the optimal parameter combination (P, σ) as the parameters of the SVM regression model.

B. THE DBF MODEL

DBN model is used to complete the feature extraction of short-term passenger flow data. By reconstructing the data with each layer of model, a more abstract way of data expression is formed, it highlights the inherent change law of the short-term passenger flow data, which can be provide to SVM. The DBF model trains weights in two steps: unsupervised weighting and supervised finetuning.

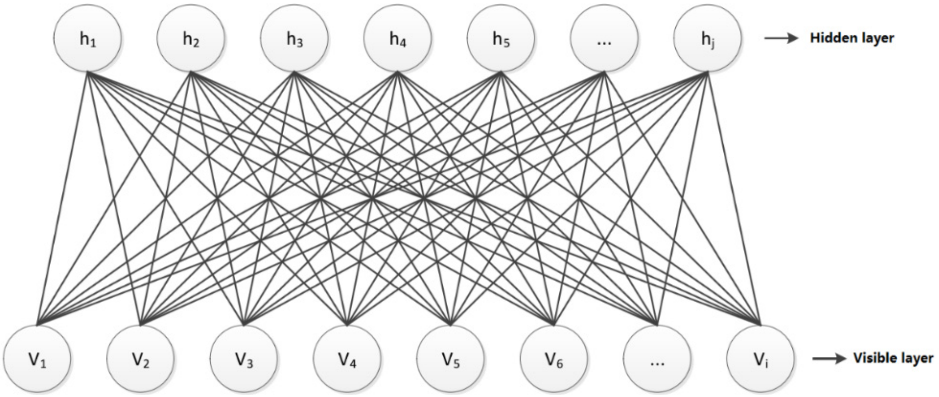


FIGURE 1. Structure of the RBM.

1) UNSUPERVISED WEIGHTING

In the first step, the underlying network of the DBN model is decomposed into multiple restricted Boltzmann machines (RBMs), and the weights are generated layer by layer to optimize the weight-eigenvector mapping on each layer.

The structure of the RBM is shown in Figure 1, where the nodes $v = \{v_1, v_2, \ldots, v_i\} \in \{0, 1\}$ belongs to the visible layer (input layer) and the nodes $h = \left\{h_1, h_2, \ldots, h_j\right\} \in \{0, 1\}$ belongs to the hidden layer. Then, the combined energy of v_i and h_j can be described as:

$$E(v, h, \vartheta) = -\sum\nolimits_{i,j} w_{ij} v_i h_j - \sum\nolimits_{i} a_i v_i - \sum\nolimits_{j} b_j h_j \quad (5)$$

where, $\vartheta = \{w_{ij}, a_i, b_j\}$ is connection weight and bias in the RBM. Then, the joint probability distribution of visible layer nodes and hidden layer nodes can be obtained as:

$$P_\theta(v, h) = \frac{1}{Z(\vartheta)} \exp(-E(v, h, \vartheta)) \quad (6)$$

where, $Z(\vartheta)$ is the normalization factor. If the visible layer nodes are known, then the probability that the j-th node in the hidden layer is 0 or 1 can be obtained as.

$$P\left(h_{j=1} \mid v\right) = \frac{1}{1 + \exp(\sum_j w_{ij} v_i - b_j)}$$

Similarly, if the hidden layer nodes are known, then the probability that the *i*-th node in the visible layer is 0 or 1 can be obtained as:

$$P(v_{i=1} \mid h) = \frac{1}{1 + \exp(\sum_j w_{ij} h_j - a_i)}$$

By Equation (6), the likelihood function $P_\vartheta(v)$ of the input v can be from the edge distribution of $P_\theta(v, h)$ to the hidden layer node h as:

$$P_\vartheta(v) = \frac{1}{Z(\vartheta)} \sum_h \exp(vwh + bh + av)$$

According to the maximum likelihood method, the maximum value of $P_\vartheta(v)$ is needed to determine the parameters of the RBM. The likelihood algorithm can be redefined as:

$$L(\vartheta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\vartheta(v^{(n)})$$

where, N represents the number of visible layer nodes.

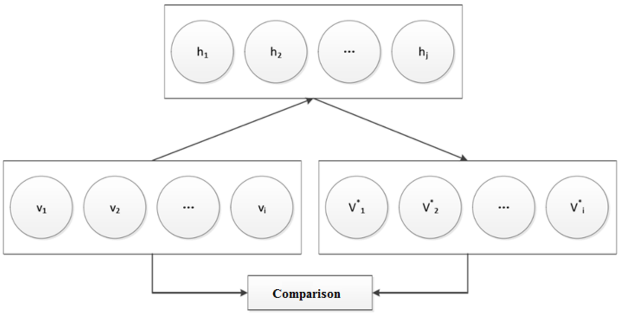


FIGURE 2. Comparison of RBM visible layer and reconstructed visible layer.

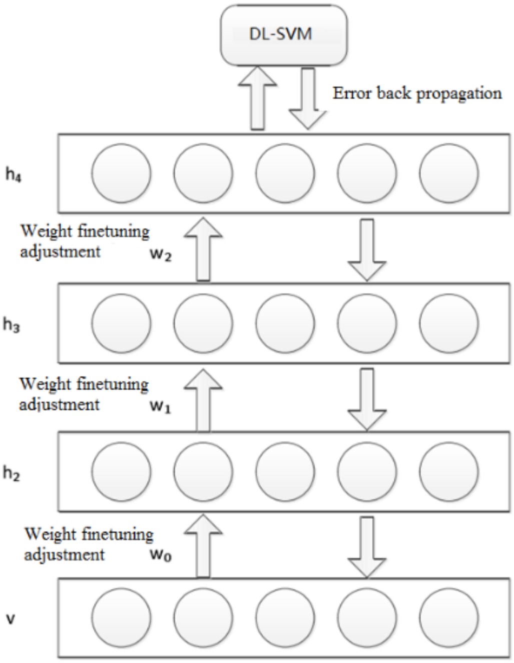
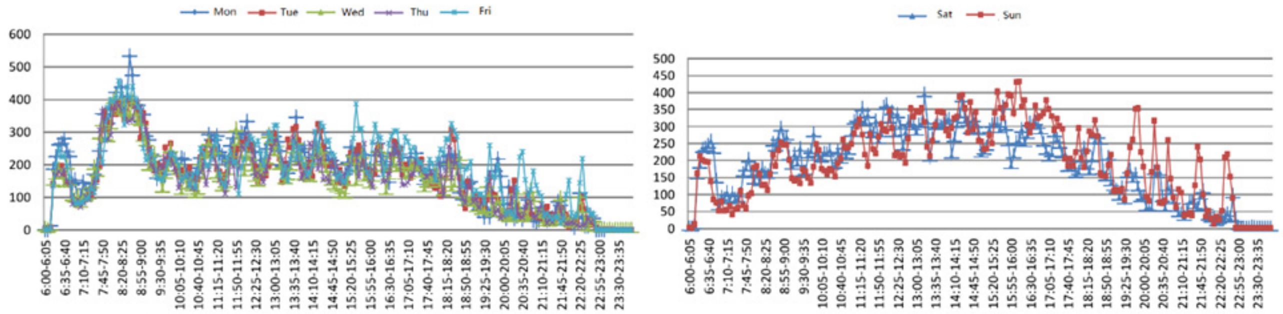

DL-SVM
Error back propagation
h4
Weight finetuning adjustment
w2
h3
Weight finetuning adjustment
w1
h2
Weight finetuning adjustment
w0
v

FIGURE 3. The flow of supervised finetuning.



(a) Workday passenger flows

(b) Weekend passenger flows

FIGURE 4. Passenger flows at 5min periods.

Since all visible and hidden layer nodes in the RBM satisfy Maxwell-Boltzmann distribution, the hidden layer h can be obtained by $P(v \mid h)$, and the visible layer v^* can be obtained by $P(v \mid h)$. The process is explained in Figure 2 below.

The unsupervised weighting can be summarized as:

Step 1: Input the sample data v into the visible layer of the RBM, and initialize the connection weight w.

Step 2: Update the state of hidden layer node h_j is updated, and compute the $P(v_i, h_j) = v_i \times h_j$ for each (v_i, h_j).

Step 3: Reconstruct v^* according to the state of hidden layer node h_j, derive h^* from v^*, and compute $P\left(v_i^*, h_j^*\right) = v_i^* \times h_j^*$.

Step 4: Update the weight of each (v_i, h_j), $w_{ij} = w_{ij} + 1 * (P(v_i, h_j) - P\left(v_i^*, h_j^*\right))$;

Step 5: Input another sample data, and repeat Steps 1∼4.

2) SUPERVISED FINETUNING

The DBN model finetunes the weights in a way like the weight training of the traditional neural network: the error between the predicted and actual values is transmitted back to each layer of the neural network, and then minimized by adjusting the weight of each layer.

In the DBN model, the matrix of short-term passenger flow data is taken as the input vector {v, target} of the DBN, and the trained neural network outputs the output vector {v*, target}. Then, the DL-SVM model predicts the short-term passenger flow v^{**} based on the v^*. The error between v^{**} and the actual value is transmitted back to the bottom layer of the DBN, and the weight of each layer is adjusted accordingly.

The supervised finetuning is illustrated in Figure 3.

IV. EXPERIMENTS AND BIG DATA ANALYSIS

Our experiments use the actual data on two-week passenger flow collected in Qingdao Metro in May, 2018. The data were divided into two parts: the workday dataset and the weekend dataset, because the workdays and weekends differ greatly in travel behavior and passenger flow trend. The daily mean incoming passenger flow was 37,529 on workdays and 40,962 on weekends. The statistical periods of passenger flow were set to 5, 10, 15 and 30min, respectively. The results of big data analysis are displayed in Figures 4∼7.

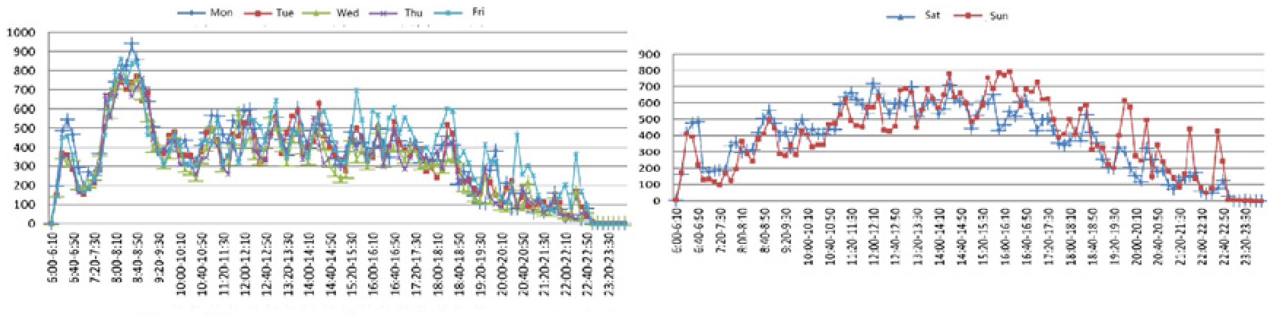
The model parameter of DL-SVM model is shown in Table 2.

On both datasets, the passenger flow curves were increasingly smooth, with the increase of statistical period, indicating that metro passenger flows within small time intervals are highly stochastic. It can also be seen that metro passenger flow is mainly influenced by longitudinal randomness and lateral periodicity.

Let $\{A_{ij}$ be the predicted metro passenger flow, $A_{i-1,j}, A_{i-2,j}, \ldots, A_{i-n,j}\}$ be the metro passenger flows in the same period of the previous n days, and $\{A_{i,j-1}, A_{i,j-2}, \ldots, A_{i,j-n}\}$ be the metro passenger flows in the first m periods of the current day. The latter two sets respectively describe the periodic effect and stochastic effect of metro passenger flow. On this basis, the sample data for DL-SVM model prediction can be expressed as eq., shown at the bottom of this page:

Taking the 5m period dataset as an example, the data were collected consecutively for 18h (6:00∼24:00) each day. In each hour, 12 groups of data were recorded. The data on one of the days were too sparse to use, and were

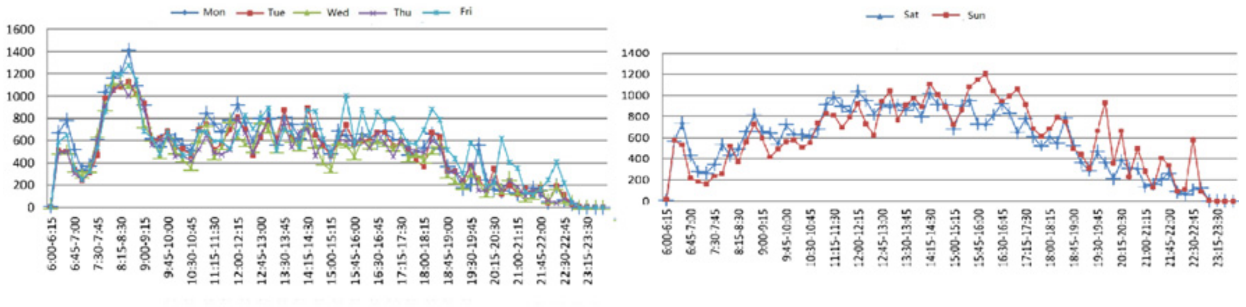
$$\begin{bmatrix} A_{i-n,j-m} & \cdots & A_{i-2,j-m} & A_{i-1,j-m} & A_{i,j-m-m} & \cdots & A_{i,j-m-1} & A_{i,j-m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ A_{i-n,j-2} & \cdots & A_{i-2,j-2} & A_{i-1,j-2} & A_{i,j-m-2} & \cdots & A_{i,j-3} & A_{i,j-2} \\ A_{i-n,j-1} & \cdots & A_{i-2,j-1} & A_{i-1,j-1} & A_{i,j-m-1} & \cdots & A_{i,j-2} & A_{i,j-1} \\ A_{i-n,j} & \cdots & A_{i-2,j} & A_{i-1,j} & A_{i,j-m} & \cdots & A_{i,j-1} & A_{i,j} \end{bmatrix}$$



(a) Workday passenger flows

(b) Weekend passenger flows

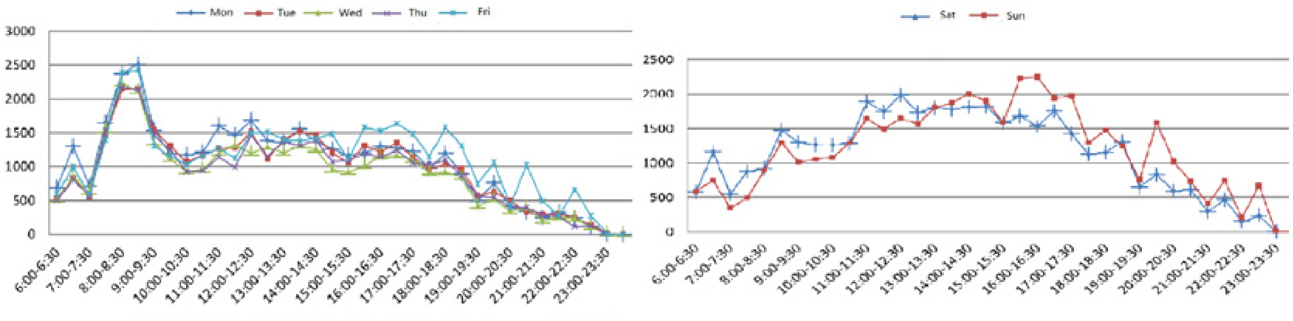
FIGURE 5. Passenger flows at 10min periods.



(a) Workday passenger flows

(b) Weekend passenger flows

FIGURE 6. Passenger flows at 15min periods.



(a) Workday passenger flows

(b) Weekend passenger flows

FIGURE 7. Passenger flows at 30min periods.

thus eliminated. The remaining 190 groups of data were randomly disturbed to avoid overconcentration in a specific period. The first 170 groups of data were taken as training samples, and the last 20 groups as test samples. Through the experiments, the DL-SVM model was compared with three common prediction models, namely, the particle swarm optimization (PSO)-SVM, the GA-SVM and the BPNN. The experimental results are displayed in Figure 8 below.

As shown in Figure 8, all models managed to predict the change law of short-term passenger flow, with a small deviation from the actual value.

Then, the performance of each model was evaluated by four indices: the MSE, the MAE, root mean square error (RMSE) and mean absolute percentage error (MAPE) [21]. The evaluation results are given in Table 3.

It can be seen that the DL-SVM outperformed all three models, indicating that this model can be applied to predict the short-term passenger flow in actual metros. It shows that DL-SVM model based on depth structure has strong algorithm performance in data sets with strong randomness and large data volume. Based on the principle of in-depth learning, the model constructs a deeper prediction model, extracts the characteristics of data by increasing the depth of network, and makes the inherent law of data more obvious.

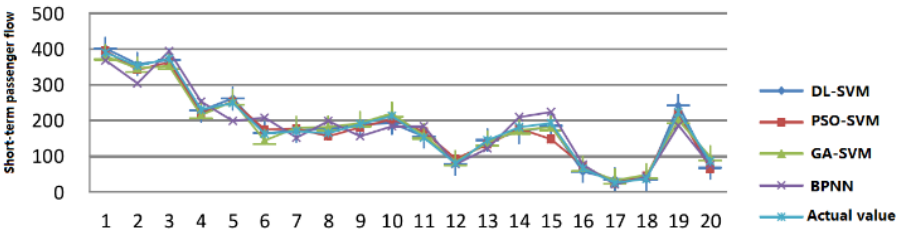


FIGURE 8. The experimental results.

TABLE 2. The model parameter of DL-SVM.

	Workday Data Set	Weekend Data Set
Number of hidden layers	4	5
Number of hidden layer nodes	12	12
Optimization algorithm	GA	GA
Kernel function	RBF kernel function	Polynomial kernel function
Learning rate	1	1
Activation function	Sigm function	Sigm function

TABLE 3. The model performance.

	MSE	RMSE	MAE	MAPE
DL-SVM	67.92	8.32	6.89	5.78 %
PSO-SVM	102.45	10.92	9.72	6.98 %
GA-SVM	136.95	11.72	10.09	7.19 %
BPNN	329.82	19.86	16.82	12.93 %

V. CONCLUSION

1) The proposed DL-SVM model enjoys excellent prediction accuracy, and provides accurate data for the operations of the URT system.
2) Through big data analysis, the trends of workday and weekend passenger flows were derived from the historical data, making it possible to know the passenger flow in the upcoming hours.
3) The intrinsic variation of the short-term passenger flow data lays a good basis for model training.
4) The DL-SVM model can effectively extract features from the short-term passenger flow data of the URT, thanks to the deep network structure and big data analysis. The feature extraction suppresses the noise in the data, and promises a good prediction performance.
5) Starting from the practical problems of metro operation, this paper mainly studies the prediction of short-term passenger flow. The model prediction accuracy threshold is improved by expanding the statistical time interval.

REFERENCES

[1] B. Xia, F. Y. Kong, and S. Y. Xie, ''Passenger flow forecast of urban rail transit based on support vector regression,'' *Appl. Mech. Mater.*, vol. 433, pp. 612–616, Mar. 2013.

[2] C. H. Zhang, R. Song, and Y. Sun, ''Kalman filter-based short-term passenger flow forecasting on bus stop,'' *J. Transp. Syst. Eng. Inf. Technol.*, vol. 11, no. 4, pp. 154–159, Jan. 2011.

[3] S. V. Kumar and L. Vanajakshi, ''Short-term traffic flow prediction using seasonal ARIMA model with limited input data,'' *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 21–29, Sep. 2015.

[4] H. Ye, Y. Hao, and H. Zhu, ''''N-Day' average volume based analysis and forecasting for daily passenger flow of Shanghai URT,'' in *Proc. Int. Conf. Transp. Eng.*, 2009, pp. 4080–4085.

[5] J. Haworth, J. Shawe-Taylor, T. Cheng, and J. Wang, ''Local online kernel ridge regression for forecasting of urban travel times,'' *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 151–178, Sep. 2014.

[6] Y. Wang, D. Zheng, S. M. Luo, D. M. Zhan, and P. Nie, ''The research of railway passenger flow prediction model based on BP neural network,'' *Adv. Mater. Res.*, vols. 605–607, pp. 2366–2369, Dec. 2013.

[7] Y. Sun, B. Leng, and W. Guan, ''A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system,'' *Neurocomputing*, vol. 166, pp. 109–121, Oct. 2015.

[8] L. Lin, Y. Li, and A. Sadek, ''A k nearest neighbor based local linear wavelet neural network model for On-line short-term traffic volume prediction,'' *Proc. Social Behav. Sci.*, vol. 96, pp. 2066–2077, Nov. 2013.

[9] X. Jiang, L. Zhang, and X. Chen, ''Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China,'' *Transp. Res. C, Emerg. Technol.*, vol. 44, no. 4, pp. 110–127, Jul. 2014.

[10] M. Li, Y. Wang, and L. Jia, ''The modeling of attraction characteristics regarding passenger flow in urban rail transit network based on field theory,'' *PLoS ONE*, vol. 12, no. 9, Sep. 2017, Art. no. e0184131.

[11] Y. Hou, P. Edara, and C. Sun, ''Traffic flow forecasting for urban work zones,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1761–1770, Aug. 2015.

[12] Y. Wei and M.-C. Chen, ''Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks,'' *Transp. Res. C, Emerg. Technol.*, vol. 21, no. 1, pp. 148–162, 2012.

[13] M.-C. Chen and Y. Wei, ''Exploring time variants for short-term passenger flow,'' *J. Transp. Geography*, vol. 19, no. 4, pp. 488–498, Jul. 2011.

[14] J. Yang and Z. Hou, ''A wavelet analysis based LS-SVM rail transit passenger flow prediction method,'' *China Railway Sci.*, vol. 34, no. 3, pp. 122–127, 2013.

[15] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, ''Traffic flow prediction with big data: A deep learning approach,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[16] W. Huang, G. Song, H. Hong, and K. Xie, ''Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[17] S. C. H. Somashekhara, A. K. Y. Setty, S. M. Sridharmurthy, P. Adiga, U. S. Mahabaleshwar, and G. Lorenzini, ''Makespan reduction using dynamic job sequencing combined with buffer optimization applying genetic algorithm in a manufacturing system,'' *Math. Model. Eng. Problems*, vol. 6, no. 1, pp. 29–37, 2019.

[18] G. Sun and S. Bin, ''Router-level Internet topology evolution model based on multi-subnet composited complex network model,'' *J. Internet Technol.*, vol. 18, no. 6, pp. 1275–1283, 2017.

[19] A. P. Gopi, N. V. Lakshman, and K. N. Ashok, ''Dynamic load balancing for client server assignment in distributed system using genetical gorithm,'' *Ingenierie Des Syst. D'Inf.*, vol. 23, no. 6, pp. 87–98, 2018.

[20] H. R. Moslemi and M. M. Keshtkar, "Sensitivity analysis and thermal performance optimization of evacuated U-tube solar collector using genetic algorithm," *Int. J. Heat Technol.*, vol. 36, no. 4, pp. 1193–1202, 2018.

[21] G. Sun and S. Bin, "A new opinion leaders detecting algorithm in multi-relationship online social networks," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4295–4307, 2018.

KAER ZHU received the bachelor's degree in engineering from the Logistical Engineering University of PLA, in 2016. He is currently pursuing the master's degree in architecture with the Army Logistics University of PLA, with a focus on planning and design of the camp. His research interests include external space design of the camp and urban planning.

PING XUN received the master's degree in architecture from Tongji University. He is currently a Professor with the Army Logistics University of PLA, with a focus on planning and design of the camp. He is one of the China Science and Technology Award Evaluation Experts, the National Natural Science Foundation Evaluation Experts, the Chongqing Landscape Committee Expert Members, and the Chongqing Planning Bureau Evaluation Experts.

WEI LI received the master's degree in civil engineering from the PLA University of Science and Technology, in 2015. She is currently a Lecturer with the Army Logistics University of PLA, with a focus on underground engineering planning and underground architectural design. She has participated in several research programs funded by Military Logistics Equipment Fund and Provincial Government.

ZHEN LI received the Ph.D. degree in architecture from Chongqing University. She is currently an Associate Professor with the Army Logistics University of PLA, with a focus on architectural history and theory and army architecture planning and design.

RUOCHONG ZHOU received the bachelor's degree in national defense architecture design from the Logistical Engineering University of PLA, in 2017. He is currently pursuing the master's degree in architecture with the Department of Military Installations, Army Logistics University of PLA. He mainly studies in the direction of military building safety.