WILEY | Hindawi

Research Article

Short-Term Traffic Flow Prediction with Weather Conditions: Based on Deep Learning Algorithms and Data Fusion

Yue Hou (ID)**, Zhiyuan Deng** (ID)**, and Hanke Cui** (ID)

School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

Correspondence should be addressed to Yue Hou; houyue@mail.lzjtu.cn

Short-term traffic flow prediction is an effective means for intelligent transportation system (ITS) to mitigate traffic congestion. However, traffic flow data with temporal features and periodic characteristics are vulnerable to weather effects, making short-term traffic flow prediction a challenging issue. However, the existing models do not consider the influence of weather changes on traffic flow, leading to poor performance under some extreme conditions. In view of the rich features of traffic data and the characteristic of being vulnerable to external weather conditions, the prediction model based on traffic data has certain limitations, so it is necessary to conduct research studies on traffic flow prediction driven by both the traffic data and weather data. This paper proposes a combined framework of stacked autoencoder (SAE) and radial basis function (RBF) neural network to predict traffic flow, which can effectively capture the temporal correlation and periodicity of traffic flow data and disturbance of weather factors. Firstly, SAE is used to process the traffic flow data in multiple time slices to acquire a preliminary prediction. Then, RBF is used to capture the relation between weather disturbance and periodicity of traffic flow so as to gain another prediction. Finally, another RBF is used for the fusion of the above two predictions on decision level, obtaining a reconstructed prediction with higher accuracy. The effectiveness and robustness of the proposed model are verified by experiments.

1. Introduction

In recent years, with the continuous increase of vehicle ownership, the conflict between road resources and travel demand is becoming increasingly acute, which leads to the increasingly serious traffic congestion and even hinders the development of social economy. ITS is an effective means to alleviate traffic congestion, and short-term traffic flow prediction is the key to it. Accurate and timely prediction of traffic flow provides reliable basis of traffic control for governors and meanwhile offers appropriate travel advice for travelers so as to optimize road network and reduce traffic congestion. However, traffic prediction is a sophisticated and nonlinear problem. Traffic flow in reality has obvious temporal correlation and periodicity, but it may evolve in an irregular way under disturbance of weather changes, which makes this problem more challenging.

The existing short-term traffic flow prediction models can be mainly divided into 3 categories: statistical models, traditional machine learning models, and deep learning models.

Statistical models include historical average (HA) and autoregressive integrated moving average (ARIMA). The former takes the statistical average value at a certain time slip in the past as the predicted value, while the latter establishes mathematical model based on the time series. This kind of method has been widely used for a long time because it can reveal the periodic changes of traffic flow data. In 1970s, Ahmed et al. [1] firstly applied ARIMA on short-term traffic flow prediction problem. After that, some improvements had been made on the ARIMA model. Voort et al. [2] combined Kohonen maps with ARIMA and proposed the KARIMA method to forecast traffic flow. Williams et al. [3] proposed seasonal ARIMA for traffic flow prediction on expressway. Min et al. [4] proposed GSTARIMA for short-term traffic flow prediction in urban network. For further extraction on spatiotemporal correlation, Duan et al. [5] proposed an extended space-time ARIMA for short-term

traffic flow estimation. However, statistical models use empirical data for parameter calculation based on transcendental knowledge, which is not suitable to reveal the nonlinearity and uncertainty of traffic flow.

Compared with the statistical methods, traditional machine learning methods like support vector machine (SVM) and support vector regression machine (SVR) show stronger function fitting ability in complex and nonlinear traffic flow prediction problem. The essential idea of this kind of method is to transform low-dimensional and linearly inseparable traffic data into high-dimensional and linearly separable expression through kernel function. Hong et al. [6] proposed a SVR traffic flow prediction model employing the hybrid genetic algorithm to determine the suitable combination of parameters. Lou et al. [7] presented the least square SVR algorithm for short-term traffic flow forecasting. Hu et al. [8] used particle swarm optimization (PSO) to determine optimal parameters for SVR for higher precision in short-term traffic flow forecasting problem. Ling et al. [9] proposed multikernel SVM and used adaptive particle swarm optimization (APSO) to improve it. Feng et al. [10] proposed a novel SVM with adaptive multikernel (AMSVM). Although there are a lot of optimization studies on this kind of method, limitation in the regression problem and lack of the ability of knowledge mining for large-scale traffic data still constrain the prediction performance.

With the emergence of traffic big data [11], short-term traffic prediction becomes more challenging and complex, which put forward higher requirements for data modeling. Deep learning models, with the effectiveness for high-dimensional space modeling and the ability to extract features of parameters through hierarchical representation, have become the mainstream technology of traffic flow prediction. Deep artificial neural network (ANN) [12], deep belief network (DBN) [13, 14] based on restricted Boltzmann machine, and long short-term memory (LSTM) network [15] for time series problems have been studied and applied to some extent. In addition, Xiao et al. [16] presented a short-term multistep freeway traffic flow prediction model with RBF whose center position of the hidden layer is determined by the fuzzy c-means clustering algorithm. Lv et al. [17] firstly used stacked autoencoder to learn the representation of traffic flow features for prediction. Abdi et al. [18] proposed a novel temporal difference backpropagation (TDBP) method in the training of RBF, which improved the short-term traffic flow prediction accuracy. Dai et al. [19] combined the spatiotemporal analysis with GRU to predict short-term traffic flow.

However, single model still has the limitation on the process of complex data. In order to integrate the advantages of single model to achieve more accurate traffic flow forecasting, a variety of combined models have emerged. Hong et al. [20] proposed the ARIMA-ANN combination model, using ARIMA to deal with the linear part of the historical data and ANN for the nonlinear part. Li et al. [21] combined the ARIMA model with the RBF model to capture the different aspects of the underlying patterns of traffic flow. Du et al. [22] proposed a hybrid deep learning framework based on RNN and CNN, which can capture the spatiotemporal dependencies of the traffic flow.

The existing short-term traffic flow prediction methods mainly aim at data modeling for traffic flow, and little research has been done on the effect of external conditions like weather on traffic flow. Hall et al. [23] discussed how adverse weather affects traffic flow. Holdener et al. [24] considered the effects of weather condition on rural freeway flow. Jian et al. [25] investigated the microscopic traffic flow parameters under rainy environment. These research studies did reveal part of the correlation between traffic flow and weather conditions, but the conclusions have not been applied to the prediction problem. Koesdwiady et al. [26] incorporated DBNs for more accurate prediction based on flow data and traffic data, and decision-level data fusion of traffic flow and weather data had been realized. Zheng et al. [27] proposed a combined architecture of embedded components, LSTM and CNN, to capture the relationship between traffic flow and weather. However, Koesdwiady et al. [26] did not have a further consideration on the weather decision (traffic prediction based on weather data only), making it hard to achieve high performance in decision-level data fusion. Reference [27] used embedding components to extract the weather disturbance but lacked the analysis and processing of weather parameters.

Based on the deficiency of existing methods, research studies on traffic flow prediction driven by both the traffic data and weather data are very important for mining data characteristics of traffic flow and improving the accuracy in prediction. This paper proposes a novel combined framework of SAE and RBF based on traffic flow and weather data. The main contributions are as follows:

(1) Corresponding data processing according to the characteristics of the data: in terms of the non-numerical weather type parameter, we firstly use one-hot coding for the original expression. Then, with an embedding component, the explicable expression is learned. To deal with numerous weather parameters, the Pearson correlation coefficient (PCC) is calculated to find out the flow-related parameters and with principal component analysis (PCA), the selected parameters are processed to be a new parameter with higher correlation. In addition, to integrate time periodicity into prediction, HA is used to construct time expression based on historical traffic flow data.

(2) Incorporating the SAE and RBF to capture the features of traffic flow and weather conditions: considering the effectiveness of combination modeling based on deep learning, we use SAE to learn the temporal correlation in traffic flow, RBF to learn the periodic evolution under weather disturbance, and another RBF to realize the decision-level data fusion of the former models. This combined framework can effectively learn the periodicity and temporal correlation of traffic flow and the disturbance of weather conditions so as to improve the accuracy and robustness of the prediction model.

2. Problem Description

Traffic prediction in our research is based on the former parameters of 12 consecutive 5-minute intervals to predict the output flow in any subsequent time slice. The output target y of the prediction model can be expressed by the following formula:

$$y = f(X_1, X_2, X_3, \ldots, X_{10}, X_{11}, X_{12}). \tag{1}$$

Considering that the evolution of traffic flow is not only restricted by its own regularity but also disturbed by external weather conditions, the input parameters of the model need to include external weather factors in addition. X_i represents the dataset on the time slice i, including flow x_i^{flow}, time expression x_i^{timecode}, vector representation of embedding $x_i^{\text{embedding}}$, and weather parameters $x_i^{\text{weathercondition}}$; X_i can be expressed as

$$X_i = \left[x_i^{\text{flow}}, x_i^{\text{timecode}}, x_i^{\text{embedding}}, x_i^{\text{weathercondition}} \right], \tag{2}$$

where y^{flow} represents the flow prediction based on traffic sequence and $y^{\text{weather\&time}}$ represents the flow prediction based on weather and time periodicity. From the perspective of decision-level data fusion, the final flow prediction value is the fusion value of two decisions, so the output y of the combined model can also be expressed as follows:

$$y = f_{\text{fusion}}\left(y^{\text{flow}}, y^{\text{weather\&time}}\right). \tag{3}$$

In multistep prediction, y is represented by y_i, and i is the step size. As shown in Figure 1(a), in the sequence-to-sequence framework, the former output is also the input of the next unit, and in this way, the prediction results can be extended to any one time slice. At the beginning, this framework was proposed for machine translation [28]. As shown in Figure 1(b), the proposed model in this paper referred to sequence-to-sequence model for multistep prediction. Considering that the input of the prediction model includes parameters of different types (weather and traffic flow), some of the modules are adjusted to satisfy the modeling demand of multivariate data.

3. Data Process

The data from January 12, 2018, to June 11, 2018, are selected as the training set, and the data from June 17, 2018, to January 12, 2019, are selected as the test set. The primary objective of data mining in our research is to study the data rules from 6:00 to 21:00, which is the peak period of the day.

3.1. Traffic Data. The traffic dataset of metro freeways in the Twin Cities is from the Regional Transportation Management Center (https://www.d.umn.edu/tdrl/traffic/). The original data are collected at a 30-second interval from more than 4,500 loop detectors. The No. 644 detector data from January 12, 2018, to January 12, 2019, with less errors and omissions are selected. In the data preprocessing stage, the data are processed into a table with 5-minute interval. Meanwhile, the omissions and error are corrected by using the principle of time similarity. Part of the processed traffic data is shown in Table 1.

To reveal the periodicity of traffic data under weather disturbance, we construct time-flow correlation expression by the HA method. The training set is divided into working days and nonworking days, and the average flow in every certain time slice is counted, which is taken as the representation of time slice. The time-flow correlation expression on time slice can be expressed as follows:

$$x_t^{\text{timecode}} = \frac{1}{n} \sum_{j=0}^{n} x_{i,j}^{\text{flow}}, \tag{4}$$

where $x_{i,j}^{\text{flow}}$ represents the flow of time slice i on day j.

3.2. Weather Data. The weather dataset is from the National Oceanic and Atmospheric Administration (https://gis.ncdc.noaa.gov/maps/ncei/lcd). On the map, we select the site with the closest location to detector 644 and the final selection is No. 72658414927. The collection time of weather data corresponds to traffic data. After data preprocessing, part of the weather data is shown in Table 2.

Weather type is a non-numerical parameter, so we use one-hot coding for the preliminary treatment. However, the sparse representation of one-hot coding cannot reflect the correlation between weather types, resulting in the model not effectively extracting its rich features during training. To solve this problem, an embedding component has been applied to extract the expression of higher dimension of weather type. The embedding vector of weather type can be expressed by the following formula:

$$x^{\text{embedding}} = f_{\text{embedding}}\left(x^{\text{one-hot}}\right), \tag{5}$$

where $x^{\text{embedding}}$ is the trained embedding vector of weather type, while $x^{\text{one-hot}}$ is the one-hot expression; the relation between them is shown in Figure 2.

Except for weather type, there are still 7 types of weather parameters. To select the parameters relevant to traffic flow, Pearson correlation coefficient ρ as formula (6) is calculated. X and Y stand for two target variables involved in the operation. The PCC of traffic flow and weather parameters is shown in Table 3, and the corresponding heat map is shown in Figure 3.

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma X \sigma Y} = \frac{E[(X - \mu X)(Y - \mu Y)]}{\sigma X \sigma Y}. \tag{6}$$

From Table 3, hourly dew-point temperature and hourly precipitation have the minimum Pearson correlation coefficient, so only the other five parameters are retained. For further extraction of weather parameters, the mathematical PCA method is used for feature-level data fusion. By linear transformation, the original set of variables with certain correlation is reformed into a new set of independent variables to replace the original ones. By this way, the original information can be retained and the similar information can be removed. Formula (7) is the original matrix A of weather parameters:
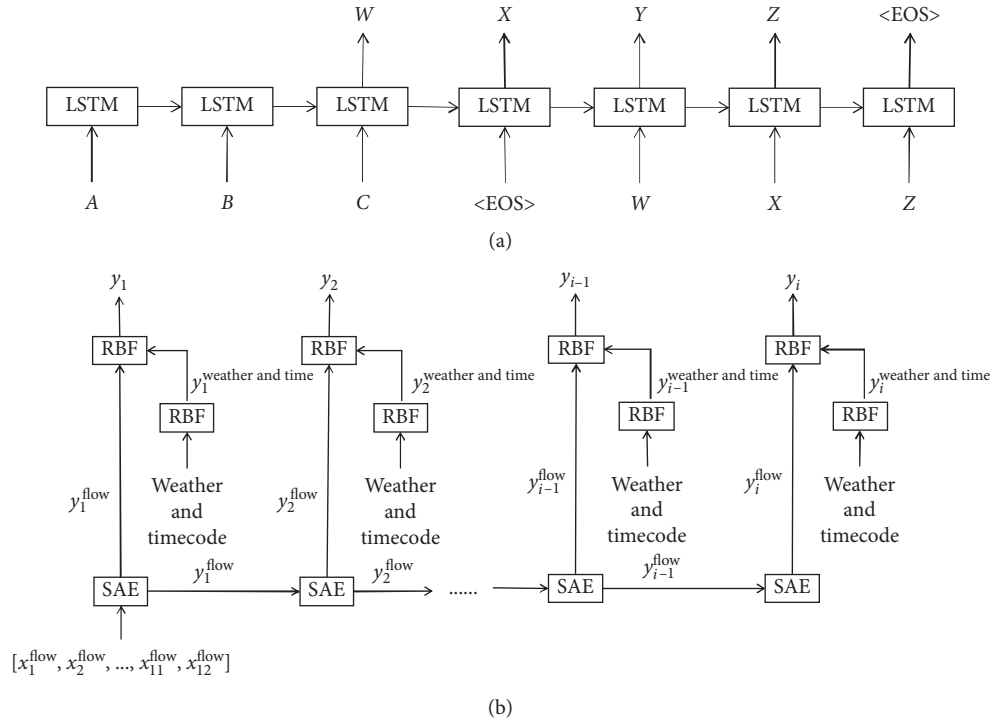
(a)

(b)

Figure 1: Multistep prediction based on sequence-to-sequence framework. (a) Sequence-to-sequence model. (b) Sequence-to-sequence proposed method.

Table 1: Traffic flow dataset (processed).

Time	Flow
2018/1/12 6:00	67
2018/1/12 6:05	86
......	......
2018/6/12 8:05	163
2018/6/12 8:10	167
......	......
2019/1/12 21:00	47

Table 2: Weather dataset (processed).

Time	Weather type	Hourly dew-point temperature	Hourly dry-bulb temperature	Hourly precipitation
......	......	......	......	......
2018/1/14 8:15	−Sn	−6	−2	0
2018/1/14 8:20	−Sn	−6	−2	0
......	......	......	......	......
2018/6/2 15:00	−RA	61	64	0.01
......	......	......	......	......

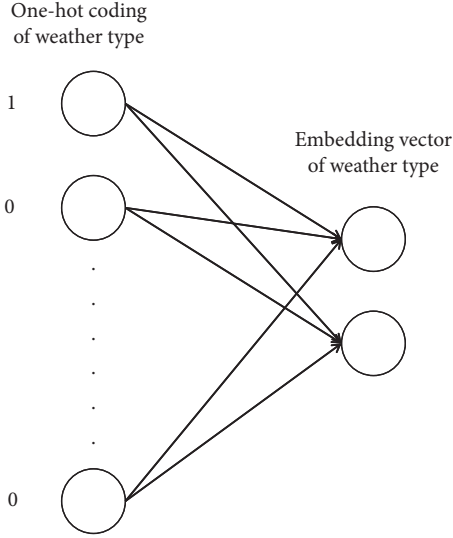| Hourly relative humidity	Hourly visibility	Hourly wet-bulb temperature	Hourly wind speed
......	......	......	......
83	6	−3	8
83	6	−3	8
......	......	......	......
88	2.5	62	3
......	......	......	......

Figure 2: Relation between one-hot coding and embedding.

Table 3: PCC of weather parameters and traffic flow.

	Flow
Hourly dew-point temperature	0.108352
Hourly dry-bulb temperature	0.234864
Hourly precipitation	−0.057385
Hourly relative humidity	−0.372337
Hourly visibility	0.143758
Hourly wet-bulb temperature	0.190240
Hourly wind speed	0.140027

$$A = \begin{pmatrix} x_1^{DB} & x_1^{RH} & x_1^{Vis} & x_1^{WB} & x_1^{WS} \\ x_2^{DB} & x_2^{RH} & x_2^{Vis} & x_2^{WB} & x_2^{WS} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{n-1}^{DB} & x_{n-1}^{RH} & x_{n-1}^{Vis} & x_{n-1}^{WB} & x_{n-1}^{WS} \\ x_n^{DB} & x_n^{RH} & x_n^{Vis} & x_n^{WB} & x_n^{WS} \end{pmatrix}, \quad (7)$$

where x^{DB}、 x^{RH}、 x^{Vis}、 x^{WB}, and x^{WS} are the selected parameters.

After being processed by PCA, the new matrix P is generated as follows:

$$P = f_{PCA}(A) = \begin{pmatrix} x_1^{pca} \\ x_2^{pca} \\ \ldots \\ x_{n-1}^{pca} \\ x_n^{pca} \end{pmatrix}, \quad (8)$$

where x^{pca} is the fusion value of the selected weather parameters processed by PCA.

4. Proposed Methodology

To capture the features of traffic flow and weather conditions, we propose a combined framework of SAE and RBF, as shown in Figure 4. The combined framework can be divided into three modules: the flow prediction module (FPM), the weather and periodicity module (WPM), and the decision-level data fusion module (DDFM).

4.1. FPM Using SAE. FPM uses SAE to extract the temporal correlation of traffic flow, as shown in Figure 5. The stacking of multiple hidden layers can improve the function fitting ability of the neural network for complicated issues. Weights of the former three hidden layers are trained by three different autoencoders (AEs) using backpropagation algorithm (BP algorithm). With these autoencoders, the input is reproduced and SAE learns multiple expressions of original data layer by layer. The input of the SAE is a sequence of flow data in continuous segments, which is denoted as

$$X^{flow} = \left[x_{i+1}^{flow}, x_{i+2}^{flow}, \ldots, x_{i+11}^{flow}, x_{i+12}^{flow} \right]. \quad (9)$$

Ignoring the calculation process of the model and focusing on the input and output, the output target of the model can be expressed as

$$y^{flow} = f_{SAE}\left(X^{flow} \right). \quad (10)$$

To train SAE, firstly, an autoencoder encodes the input X^{flow} to a sparse representation as shown in the following formula:

$$X^{hidden1} = W^{hidden1} X^{flow}, \quad (11)$$

where $X^{hidden1}$ is the learned sparse representation of the original input data and $W_{hidden1}$ is the weight matrix between the input layer and the first hidden layer.

Then, with the other two autoencoders, the input is reconstructed again, as shown in formulas (12) and (13).

$$X^{hidden2} = W^{hidden2} X^{hidden1}, \quad (12)$$

$$X^{hidden3} = W^{hidden3} X^{hidden2}, \quad (13)$$

where $W_{hidden1}$, $W_{hidden2}$, and $W_{hidden3}$ are transferred to SAE as the weights of first three layers. The weights of the last layer of the SAE are trained separately in all with BP algorithm after weight transfer. In this way, after training, SAE can capture the time correlation of traffic flow data according to the former traffic flow which is close to the predicted value in time.

4.2. WPM Using RBF. As shown in Figure 6, RBF is a three-layer neural network, consisting of an input layer, hidden layer, and output layer. Unlike general ANN, the transformation of RBF from input space to hidden layer space is nonlinear, while that from hidden layer space to output layer space is linear. The training of the RBF can be summarized with two stages: (a) determine the center of basis function of the hidden layer with unsupervised learning like clustering algorithm and (b) train weights between hidden layer and output layer with supervised learning.

The purpose of WPM is to generate the flow prediction with three processed parameters including $x^{embedding}$, $x^{timecode}$, and x^{pca}, so the function of RBF can be described as follows:
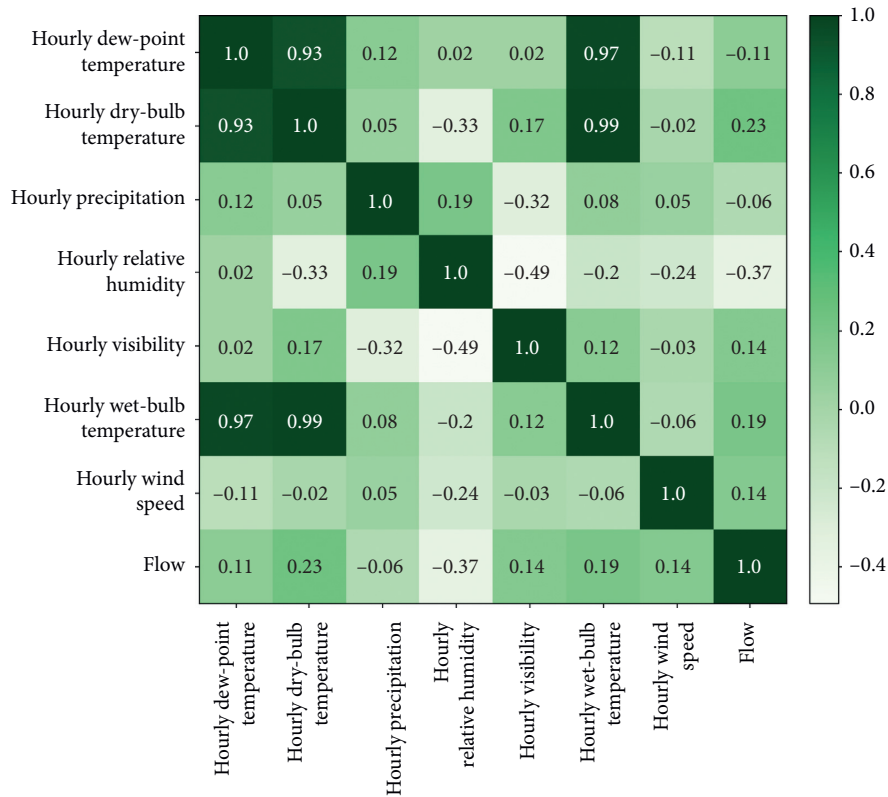
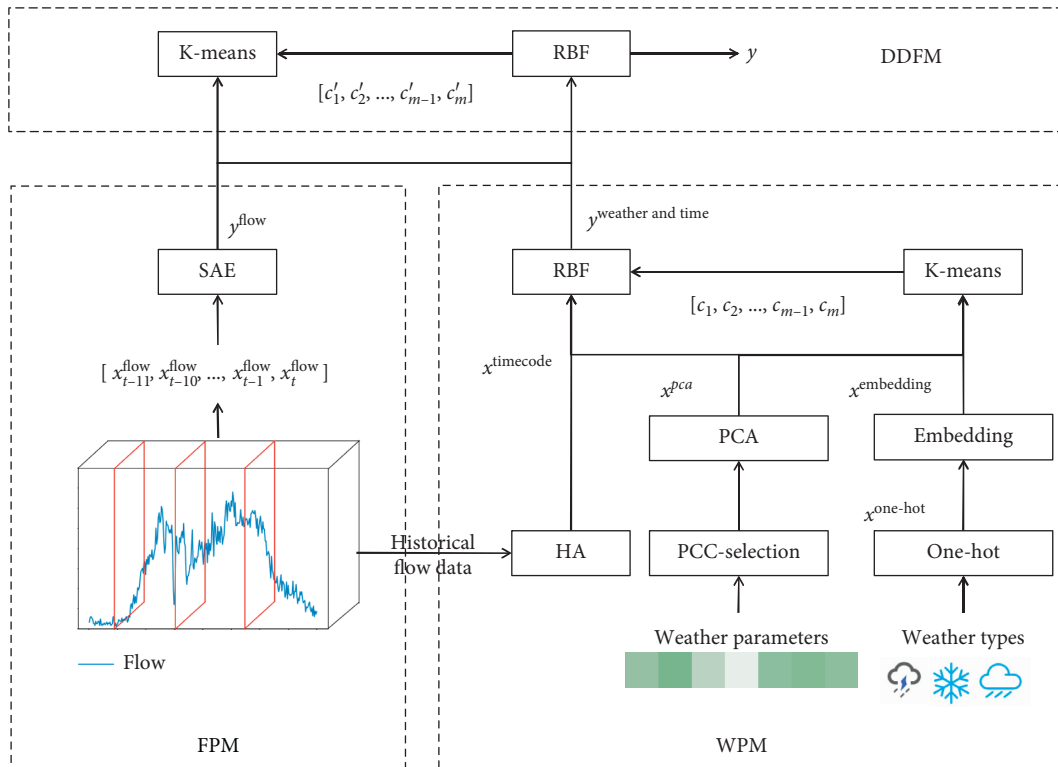Figure 3: Heat map of PCC of weather parameters and traffic flow.

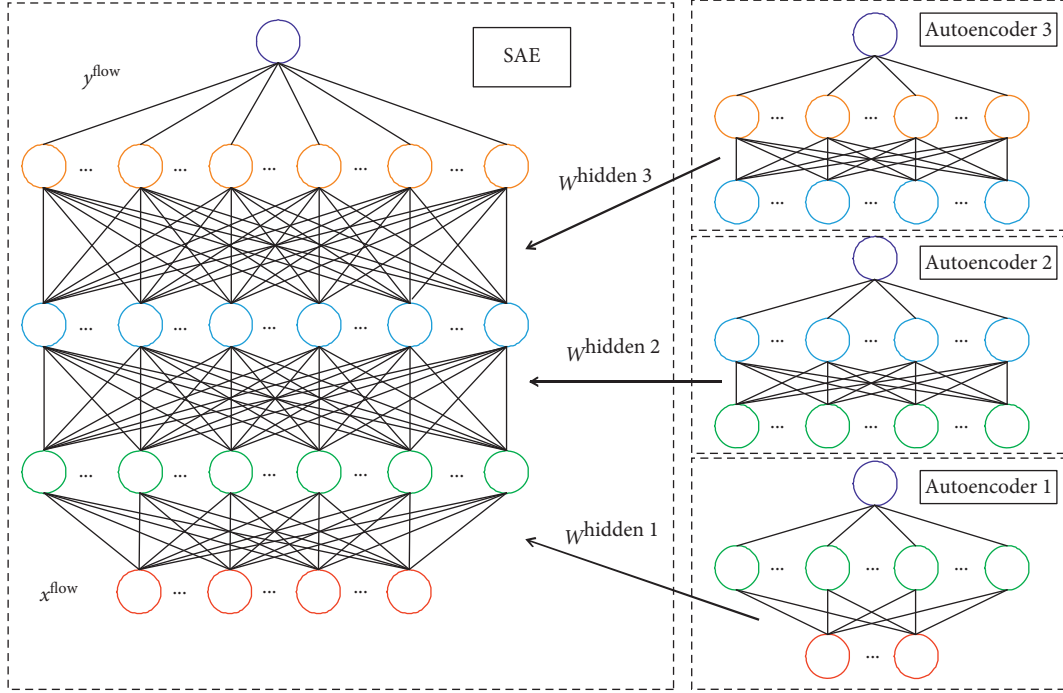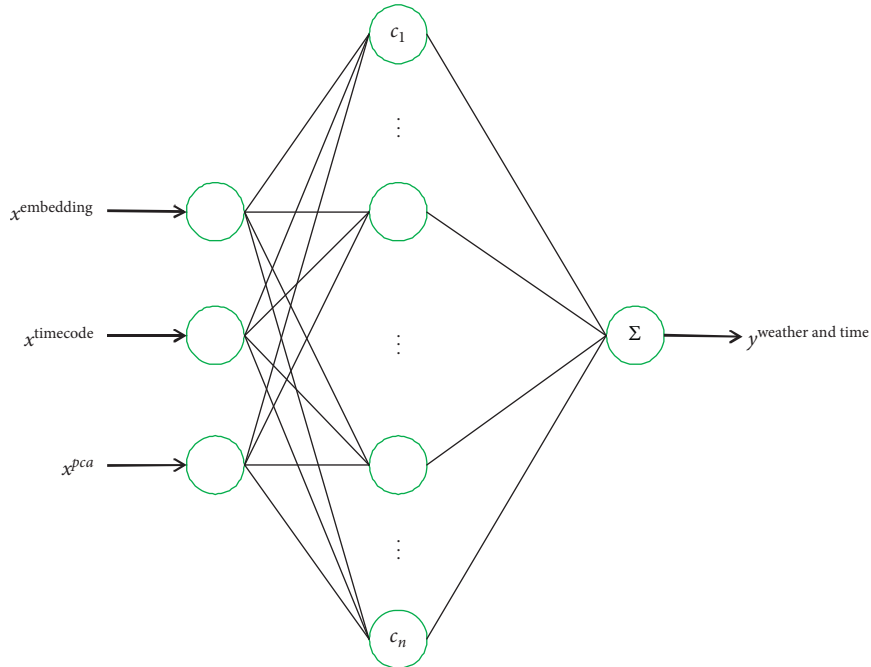Figure 4: The combined framework of SAE and RBF.

FIGURE 5: SAE structure in FPM.

FIGURE 6: The structure of RBF in WPM.

$$y^{\text{weather\&time}} = f_{\text{RBF}}\left(x^{\text{embedding}}, x^{\text{timecode}}, x^{pca}\right). \quad (14)$$

In part (a), the K-means clustering algorithm is applied to find out m cluster centers of the input data as the radial basis of Gaussian kernel function. This procedure is summarized in Algorithm 1.

The selected set of cluster centers from Algorithm 1 is also called the set of radial basis, which can be defined as

$C = [c_1, c_2, \ldots, c_{m-1}, c_m]$, where c_i represents the selected cluster center. Define $x_i = [x_i^{\text{embedding}}, x_i^{\text{timecode}}, x_i^{pca}]$ as the input sequence, and the Gaussian kernel function R is also regarded as the activation function of the network, and the definition is

$$R\left(x_i - c_j\right) = \exp\left(-\frac{1}{2\sigma^2}\big\|[\,x_i - c_j]\big\|^2\right), \quad (15)$$

Input: given the training set of $[x^{embedding}, x^{timecode}, x^{pca}]$, the number of cluster centers m

Procedure:

Step 1: m points were selected randomly as clustering centers.

Step 2: calculate the distance between every point and the m clustering centers, and then the points are divided into the nearest clustering center, thus forming m clusters.

Step 3: recalculate the centroid (mean value) of each cluster, forming m new clustering centers.

Step 4: repeat step 2 to step 4 until the cluster centers no longer change or the set number of iterations is reached.

Output: the m cluster centers in the final iteration as $C = [c_1, c_2, \ldots, c_{m-1}, c_m]$.

ALGORITHM 1: K-means clustering.

where σ represents the hyperparameters.

With formula (15), linear indivisibility in low dimension becomes linearly separable in high dimension, which is the core idea of kernel function. Thus, nonlinear mapping from input layer to hidden layer is finished. Then, with formula (16), the output target of RBF is determined.

$$y_i^{weather\&time} = \sum_{j=1}^{h} w_j R(x_i - c_j), \quad (16)$$

where w_j represents trainable weights from hidden layer to output layer.

In part (b), to determine the value of w_j, different methods of parameter updating can be attempted, including BP algorithm and pseudoinverse matrix method (PIM). PIM is a network training algorithm similar to BP, but the calculation of PIM is more simple. Since the output of the network is linear to the adjustable parameters, using PIM to solve linear equations directly for the weights is more efficient and accurate than using BP algorithm.

Pseudoinverse matrix is a generalized form of inverse matrix, aiming at the singular matrix or nonsquare matrix with no corresponding inverse matrix. Use W to represent the set of w_j, Y to represent the output of the network, and X to represent the output matrix of the hidden layer processed by Gaussian kernel activation function. In training stage, we assumed that the input and the output of RBF satisfy the following formula:

$$XW = Y, \quad (17)$$

where W can be calculated as

$$W = \text{dot}(pinv(X), Y), \quad (18)$$

where dot is vector dot multiplication and $pinv(X)$ is the pseudoinverse matrix of X. Algorithm 2 shows the calculation process of $pinv(X)$.

4.3. DDFM Using RBF. As shown in Figure 7, DDFM is realized by another RBF with the same configuration of that in WPM and differs from its input dimension. Different from the former which used RBF for data fusion on feature level, DDFM is designed for the decision-level data fusion of the outputs of the former modules. RBF is a kind of neural network with simple structure, so there is no need to consider the hierarchical structure when modeling, and it

can meet the requirement of data fusion both in feature level and decision level. The output of DDFM is also the final prediction of the combined model, and the definition is

$$y = y_{fusion} = f_{fusion}\left(y^{flow}, y^{weather\&time}\right). \quad (19)$$

5. Experimental Results

5.1. Evaluation Indexes. To test the performance of the proposed model, a variety of evaluation indexes were used in the experimental stage, including mean absolute percentage error (MAPE), symmetric mean absolute percentage error (SMAPE), mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE). These metrics mainly reflect the gap between the real values and the predicted values. The specific formulas of these metrics are as follows:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_i^{predict} - y_i^{true}}{y_i^{true}} \right|,$$

$$\text{SMAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \frac{\left| y_i^{predict} - y_i^{true} \right|}{\left(\left| y_i^{predict} \right| + \left| y_i^{true} \right| \right)/2},$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i^{predict} - y_i^{true} \right|, \quad (20)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^{predict} - y_i^{true} \right)^2,$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(y_i^{predict} - y_i^{true} \right)^2}$$

5.2. Configuration and Baselines. The proposed model and the baseline-model are implemented by Python-Keras, the data are normalized by MinMaxScaler, the iteration time is 600, the batch size is 256, the function optimizer selects RMSprop, and the loss function is MSE. According to the actual performance of each model based on the above standards, the relevant parameters are fine tuned. The used baselines include the following:
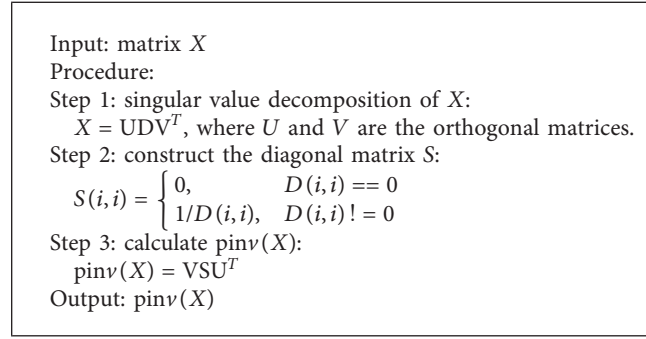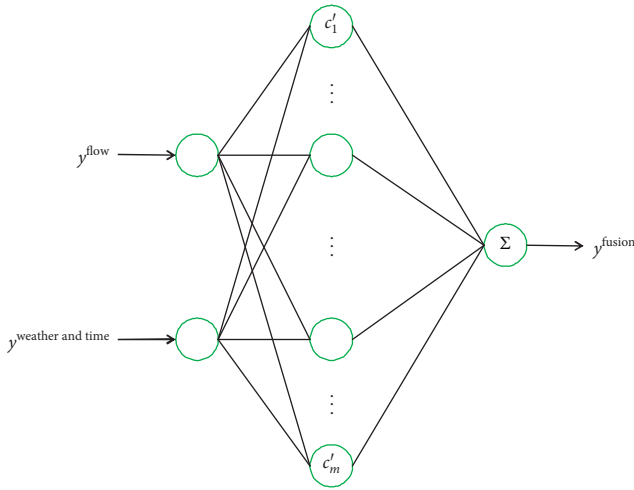
```
Input: matrix X
Procedure:
Step 1: singular value decomposition of X:
    X = UDV^T, where U and V are the orthogonal matrices.
Step 2: construct the diagonal matrix S:
    S(i, i) = { 0,         D(i, i) == 0
              { 1/D(i, i),  D(i, i) ! = 0
Step 3: calculate pinv(X):
    pinv(X) = VSU^T
Output: pinv(X)
```

ALGORITHM 2: Calculation of $\text{pin}\nu(X)$.

FIGURE 7: The structure of RBF in DDFM.

(1) HA: use historical averages as prediction value.

(2) LSTM: long short-term memory network for time series problem.

(3) GRU: a variant of LSTM with simpler structure.

(4) StackedLSTM: the stacking of LSTM layer.

(5) StackedGRU: the stacking of GRU layer.

(6) SAE: stacked autoencoders, as is described in Section 4.

(7) Fusion with ANN (F-ANN): ANN for decision-level data fusion of the outputs of FPM and WPM.

(8) Fusion with RBF on feature and decision level (F-RBF-FDL): RBF for feature and decision-level data fusion of the outputs of FPM and the processed data $x^{\text{embedding}}$, x^{timecode}, and x^{pca}.

5.3. Performance Comparison in WPM. The purpose of WPM is to capture the periodic evolution and weather disturbance. In data preprocessing stage, efforts have been done to meet the requirements of feature extraction of weather data, including PCC for feature selection and PCA for feature-level data fusion of the selected weather parameters. In addition, the scheme of WPM is selected from numerous experimental schemes. Some controlled trials with different scheme have been done. The comparison is shown in Table 4. The specific contents of each scheme are as follows:

(a) RBF + PIM: use RBF without any process on the weather parameters. Use PIM to train the network.

(b) RBF + PCC + PIM: on the basis of scheme a, use PCC to reduce the amount of weather parameters. Use PIM to train the network.

(c) RBF + PCA + PIM: on the basis of scheme a, use PCA to fuse the weather parameters. Use PIM to train the network.

(d) RBF + PCC + PCA + PIM: the fusion of scheme a and scheme b. Use PIM to train the network.

(e) RBF + PCC + PCA + BP: similar to scheme d, but use BP instead of PIM in training.

From Table 4, our selected scheme (RBF + PCC + PCA + PIM) achieves the best performance in all metrics. Control groups a–d confirm that the model with feature selection by PCC and feature-level data fusion by PCA can get more accurate prediction value. That is, our proposed method (PCC + PCA) for data process of weather parameters is significant for precise prediction by accurate extraction of periodicity and weather disturbance. Comparing experiment *d* with experiment *e*, we can conclude that using pseudoinverse matrix for solution (or weights) of linear equation in RBF can achieve higher accuracy in a simpler way and BP algorithm is probably not suitable for the training of RBF.

5.4. Performance Comparison between Proposed Model and Baselines in Single Step. The performance comparison between our proposed model and the baselines is shown in Figure 8 and Table 5. The proposed model has the optimal performance in all the selected error indexes, with 10.378414% of MAPE, 10.059485% of SMAPE, 9.494741 of MAE, 151.153208 of MSE, and 12.294438 of RMSE. In Table 5, models g, h, and i get better performance in prediction for they all consider periodicity and weather disturbance in modeling. Experiment groups h and i confirm the effectiveness of decision-level data fusion. That is, with

TABLE 4: Performance comparison in WPM.

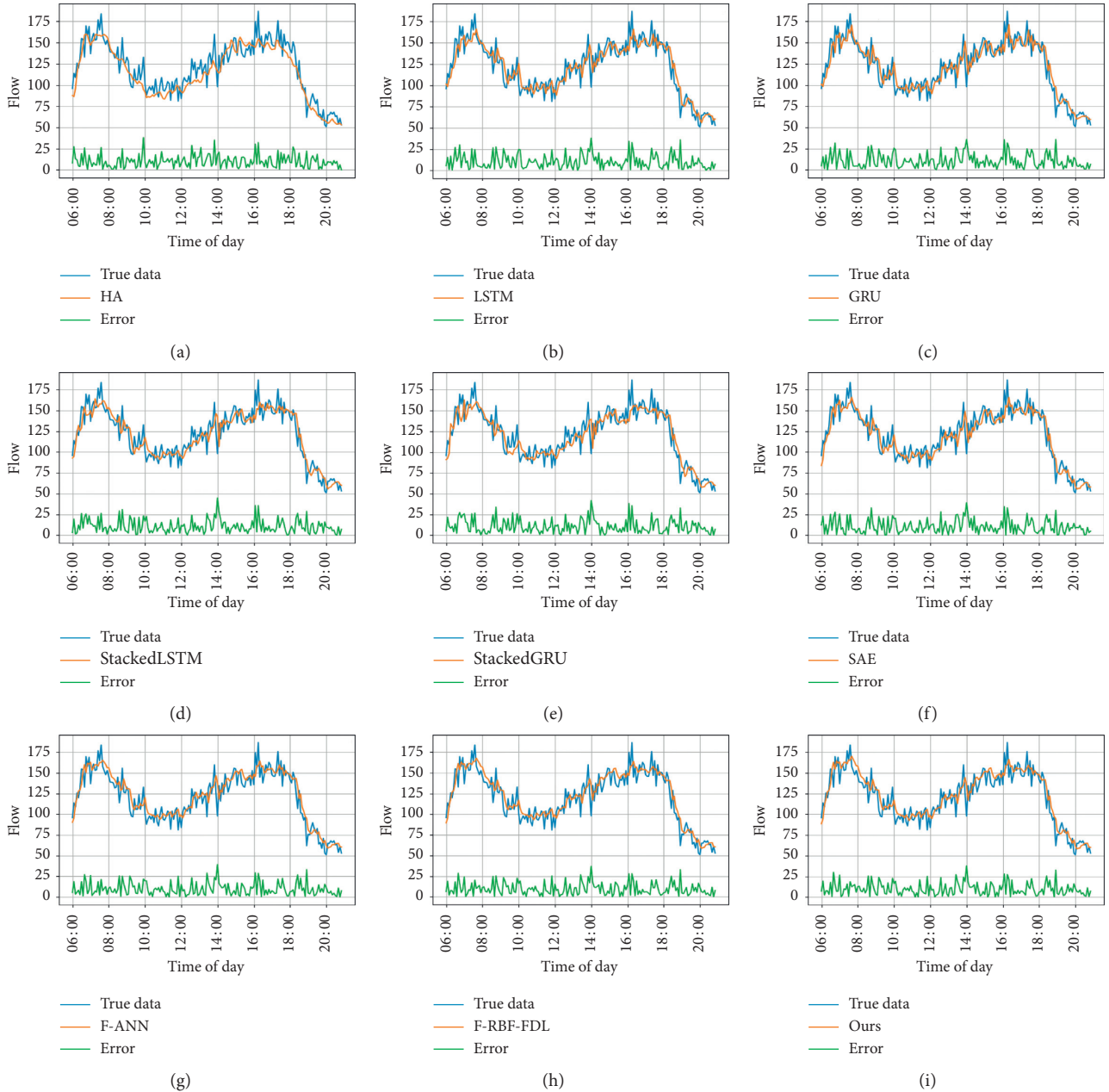| Model	MAPE (%)	MAE	MSE	RMSE
(a) RBF + PIM	20.844211	14.272642	403.534660	20.088172
(b) RBF + PCC + PIM	20.375935	14.172910	398.247865	19.956149
(c) RBF + PCA + PIM	18.955604	14.088218	389.717634	19.741267
(d) RBF + PCC + PCA + PIM	18.857257	13.988252	384.716684	19.614196
(e) RBF + PCC + PCA + BP	21.445947	17.124295	517.260612	22.743364

FIGURE 8: Performance comparison of different models. Prediction with traffic data only: (a) HA. (b) LSTM. (c) GRU. (d) StackedLSTM. (e) StackedGRU. (f) SAE. Prediction with both the traffic data and weather data: (g) F-ANN. (h) F-RBF-FDL. (i) Ours.

the combination modeling idea and selected data fusion scheme, the proposed model can effectively learn the temporal correlation in traffic flow, the time periodicity from historical data, and the disturbance of weather conditions so as to improve the accuracy and robustness. In Figure 8, we

divided these methods into two categories: prediction with traffic data only (a–f) and prediction with multivariate data (g–i). The groups g–i which consider the weather conditions show higher accuracy compared with the groups a–f which do not.

TABLE 5: Performance comparison of the proposed one and baselines.

| Model	MAPE (%)	SMAPE (%)	MAE	MSE	RMSE
(a) HA	20.511022	17.320345	15.748870	462.314322	21.501496
(b) LSTM	11.341084	10.692583	9.998004	166.770930	12.913982
(c) GRU	11.896406	10.963014	10.053730	167.202068	12.930664
(d) StackedLSTM	10.573702	10.612538	9.695501	155.607847	12.474287
(e) StackedGRU	10.628161	10.316958	9.778009	158.479036	12.588846
(f) SAE	10.558251	10.397757	9.841813	162.254182	12.737903
(g) F-ANN	10.537031	10.134355	9.511839	151.190477	12.295954
(h) F-RBF-FDL	10.521758	10.134487	9.507794	151.528273	12.309682
(i) Ours	10.378414	10.059485	9.494741	151.153208	12.294438
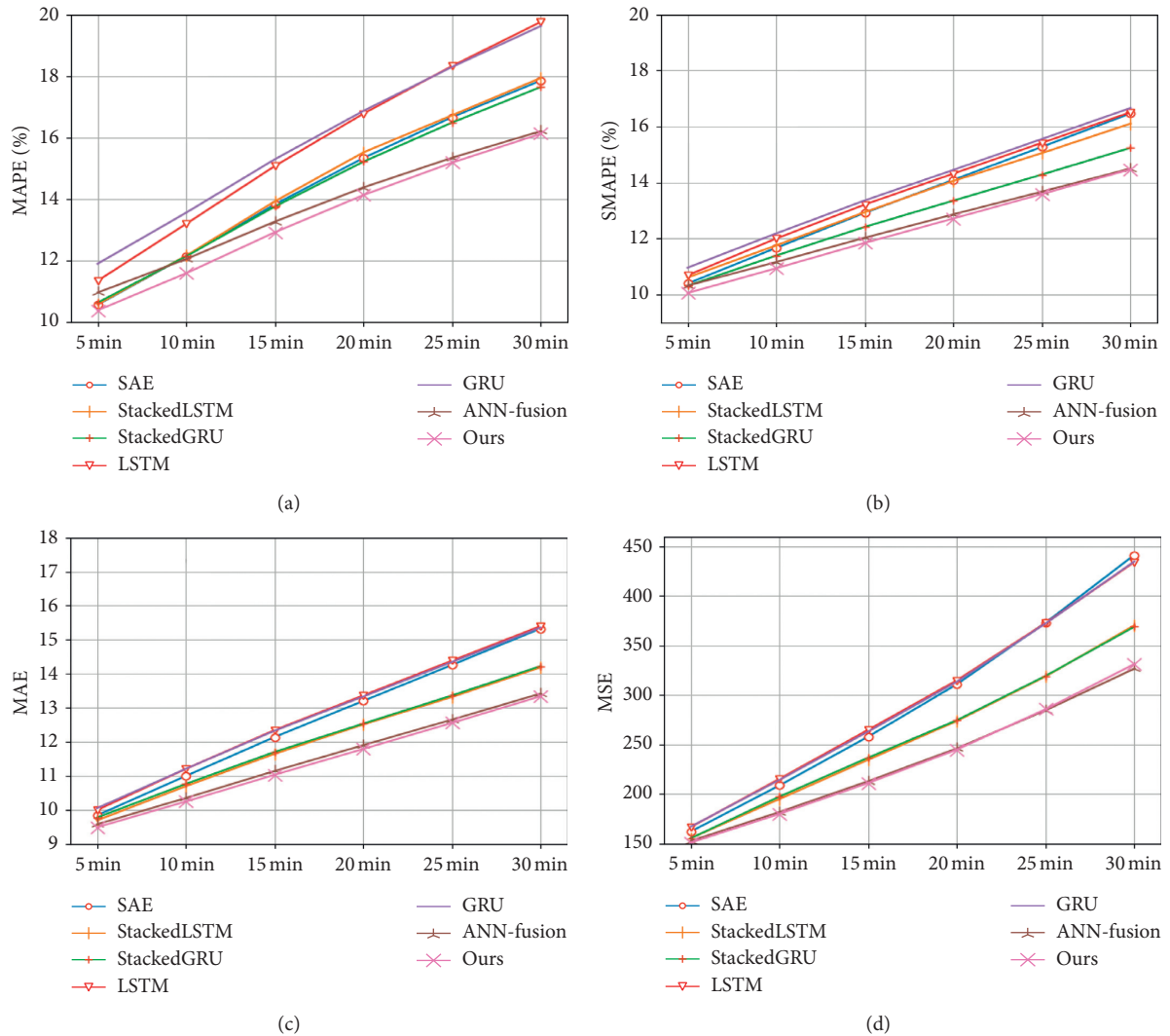
FIGURE 9: The changing indicators of models in multiple steps.

5.5. Performance Comparison between Proposed Model and Baselines in Multiple Steps.

In Figure 9, with the increase of time step size, an upward trend of prediction error occurred in all models for the accumulated error from the last step, but the proposed model is still superior to baselines in most indicators. Compared with experiments in single step, the proposed model shows more obvious advantages of prediction accuracy in multiple steps. With the increase of

prediction time span, the accuracy gap between models will be further widened, making the advantages of proposed model more obvious. From Table 6, we can see the indicators of the models in different steps. Models f and g achieve more precise prediction value when compared to the other models because they combine time periodicity with weather disturbance in modeling. Comparing f with g, g is slightly better than f in most indicators and most step sizes. In Step 05 and

TABLE 6: Performance comparison in multiple steps.

Prediction step	Model	MAPE (%)	SMAPE (%)	MAE	MSE	RMSE
Step 01 (5 min)	(a) LSTM	11.34	10.69	9.99	166.77	12.91
	(b) GRU	11.89	10.96	10.05	167.20	12.93
	(c) StackedLSTM	10.57	10.61	9.69	155.60	12.47
	(d) StackedGRU	10.62	10.31	9.77	158.47	12.58
	(e) SAE	10.55	10.39	9.84	162.25	12.73
	(f) F-ANN	10.53	10.13	9.51	151.19	12.29
	(g) Ours	10.37	10.05	9.49	151.15	12.29
Step 02 (10 min)	(a) LSTM	13.19	12.00	11.20	215.49	14.67
	(b) GRU	13.55	12.18	11.20	214.40	14.64
	(c) StackedLSTM	12.15	11.75	10.69	195.16	13.97
	(d) StackedGRU	12.14	11.39	10.76	197.55	14.05
	(e) SAE	12.15	11.67	10.99	208.83	14.45
	(f) F-ANN	12.05	11.16	10.35	182.06	13.49
	(g) Ours	11.58	10.94	10.24	180.19	13.42
Step 03 (15 min)	(a) LSTM	15.07	13.21	12.34	264.90	16.27
	(b) GRU	15.29	13.36	12.32	263.18	16.22
	(c) StackedLSTM	13.92	12.94	11.64	234.77	15.32
	(d) StackedGRU	13.75	12.41	11.70	236.78	15.38
	(e) SAE	13.82	12.91	12.14	258.00	16.06
	(f) F-ANN	13.26	12.03	11.14	213.09	14.59
	(g) Ours	12.92	11.84	11.03	211.28	14.53
Step 04 (20 min)	(a) LSTM	16.77	14.31	13.36	314.91	17.74
	(b) GRU	16.86	14.45	13.33	313.58	17.70
	(c) StackedLSTM	15.51	14.05	12.51	273.88	16.54
	(d) StackedGRU	15.21	13.35	12.53	274.65	16.57
	(e) SAE	15.33	14.09	13.20	311.01	17.63
	(f) F-ANN	14.37	12.86	11.90	246.19	15.69
	(g) Ours	14.12	12.71	11.79	244.99	15.65
Step 05 (25 min)	(a) LSTM	18.32	15.41	14.38	372.75	19.30
	(b) GRU	18.29	15.55	14.34	372.23	19.29
	(c) StackedLSTM	16.72	15.05	13.32	318.88	17.85
	(d) StackedGRU	16.48	14.28	13.36	319.37	17.87
	(e) SAE	16.66	15.27	14.26	373.09	19.31
	(f) F-ANN	15.34	13.67	12.65	284.67	16.87
	(g) Ours	15.18	13.59	12.55	285.80	16.90
Step 06 (30 min)	(a) LSTM	19.75	16.50	15.40	434.19	20.83
	(b) GRU	19.62	16.66	15.36	434.89	20.85
	(c) StackedLSTM	17.93	16.10	14.19	370.32	19.24
	(d) StackedGRU	17.64	15.23	14.22	369.00	19.20
	(e) SAE	17.85	16.46	15.33	441.44	21.01
	(f) F-ANN	16.20	14.50	13.42	326.71	18.07
	(g) Ours	16.12	14.47	13.34	331.33	18.20

Step 06, f shows slightly better performance in MSE and RMSE than g because time periodicity in modeling is more important for the prediction accuracy in long-term prediction. That is, the proposed model is more stable and precise no matter in single step or in multiple step.

6. Conclusion

A combined framework of SAE and RBF is proposed for short-term traffic flow prediction based on traffic data and weather data. Before modeling for traffic prediction, lots of work for data processing are involved in our experiment. For precise numerical representation for time periodicity, HA is used to create time expression based on historical flow. In terms of weather data, one-hot coding and embedding component are used for numerical expression of weather type while PCC and PCA are applied to the feature-level data fusion of weather parameters. After data processing, we incorporate SAE and RBF to capture the features of traffic flow and weather conditions. The final prediction considers the temporal correlation, time periodicity, and weather disturbance, owning higher accuracy and robustness. Quantities of experiments have been done to test the performance of the proposed model from different aspects,

confirming that the research of our short-term traffic flow prediction model driven by both the traffic data and weather data is valuable.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known conflicting financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transportation Research Record*, vol. 3, pp. 1–9, 1979.

[2] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[3] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[4] X. Min, J. Hu, and Z. Zuo, "Urban traffic network modeling and short-term traffic flow forecasting based on GSTARIMA model," in *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems*, pp. 1535–1540, Qingdao, China, September 2010.

[5] P. Duan, G. Mao, C. Zhang, and S. Wang, "STARIMA-based traffic prediction with time-varying lags," in *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems*, pp. 1610–1615, Qingdao, China, November 2016.

[6] W.-C. Hong, Y. Dong, F. Zheng, and S. Y. Wei, "Hybrid evolutionary algorithms in a SVR traffic flow forecasting model," *Applied Mathematics and Computation*, vol. 217, no. 15, pp. 6733–6747, 2011.

[7] W. Lou, Y. Zhou, S. Peng, and J. Wang, "An improved least square support vector regression algorithm for traffic flow forecasting," in *Proceedings of the17th International IEEE Conference on Intelligent Transportation Systems*, pp. 2379–2384, ITSC), Qingdao, China, 2014.

[8] W. Hu, L. Yan, K. Liu, and H. Wang, "PSO-SVR: a hybrid short-term traffic flow forecasting method," in *Proceedings of the 2015 IEEE 21st International Conference on Parallel and Distributed Systems*, pp. 553–561, ICPADS) VIC, Melbourne, Australia, 2015.

[9] X. Ling, X. Feng, Z. Chen, Y. Xu, and H. Zheng, "Short-term traffic flow prediction with optimized multi-kernel support vector machine," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 294–300, San Sebastian, Spain, 2017.

[10] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2019.

[11] F. Xia, A. Rahim, X. Kong, M. Wang, Y. Cai, and J. Wang, "Modeling and analysis of large-scale urban mobility for green transportation," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1469–1481, 2018.

[12] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.

[13] J. Liu, N. Wu, Y. Qiao, and Z. Li, "Short-term traffic flow forecasting using ensemble approach based on deep Belief networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, 2020.

[14] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep Belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[15] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 153–158, Chengdu, China, 2015.

[16] J.-M. Xiao and Xi-H. Wang, "Study on traffic flow prediction using RBF neural network," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 5, pp. 2672–2675, Shanghai, China, 2004.

[17] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1–9, 2014.

[18] J. Abdi, B. Moshiri, and A. K. Sedigh, "Comparison of RBF and MLP neural networks in short-term traffic flow forecasting," in *Proceedings of the 2010 International Conference on Power, Control and Embedded Systems*, pp. 1–4, Allahabad, India, 2010.

[19] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU," *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[20] H. hong-qiong and T. tian-hao, "Short-term traffic flow forecasting based on ARIMA-ANN," in *Proceedings of the 2007 IEEE International Conference on Control and Automation*, pp. 2370–2373, Guangzhou, China, 2007.

[21] K. Li, C. Zhai, and J. Xu, "Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN," in *Proceedings of the2017 Chinese Automation Congress (CAC)*, pp. 2804–2807, Jinan, China, 2017.

[22] S. Du, T. Li, X. Gong, Y. Yang, and S. J. Horng, "Traffic flow forecasting based on hybrid deep learning framework," in *Proceedings of the2017 12th International Conference on Intelligent Systems and Knowledge Engineering*, pp. 1–6, ISKE), Nanjing, China, 2017.

[23] F. L. Hall and D. Barrow, "Effect of weather on the relationship between flow and occupancy on freeways," *Transportation Research Record*, vol. 1194, pp. 55–63, 1988.

[24] M. Kyte, Z. Khatib, P. Shannon, and F. Kitchener, "Effect of weather on free-flow speed," *Transportation Research Record:*

Journal of the Transportation Research Board, vol. 1776, no. 1, pp. 60–68, 2001.

[25] W. Jian, S. Xixiong, and H. Xiaowei, "Researching on microscopic traffic flow characteristics of freeway under rainy environment," in *Proceedings of the2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications*, pp. 889–892, Hunan, 2014.

[26] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: a deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508–9517, 2016.

[27] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, and Y. Zhang, "Deep and embedded learning approach for traffic flow prediction in urban informatics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3927–3939, 2019.

[28] I. Sutskever, Y. Oriol Vinyals, and V. Quoc, "Sequence to Sequence Learning with Neural Networks," 2014, http://arxiv.org/abs/1409.3215.