IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2019, accepted May 27, 2019, date of publication May 30, 2019, date of current version June 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919996

Traffic Flow Data Prediction Using Residual Deconvolution Based Deep Generative Network

DI ZANG[1], YANG FANG[1], ZHIHUA WEI[1], KESHUANG TANG[2], AND JIUJUN CHENG[1]

[1]Key Laboratory of Embedded System and Service Computing, Department of Computer Science and Technology, Ministry of Education, Tongji University, Shanghai 200092, China

[2]Department of Traffic Information Engineering and Control, Tongji University, Shanghai 200092, China

Corresponding authors: Di Zang (zangdi@tongji.edu.cn) and Jiujun Cheng (chengjj@tongji.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876218, Grant 61872271, and Grant 61573259, and in part by the Fundamental Research Funds for the Central Universities under Grant 22120180302.

ABSTRACT Traffic flow prediction is quite crucial for estimating the future traffic states, efficient and accurate prediction models greatly contribute to the smooth traffic of road networks. However, existing methods mainly concentrate on short-term prediction. The challenging task of long-term flow prediction for the next day, as the important reference of traffic management, is still not well solved. In this paper, we present a residual deconvolution based deep generative network (RDBDGN) to handle the problem of long-term traffic flow prediction. The proposed method consists of a generator and a discriminator. The generator is composed of multi-channel residual deconvolutional neural networks, and the discriminator contains a convolutional neural network which aims to optimize the adversarial training process. The experiments are evaluated based on the traffic flow data of elevated highways, presented results demonstrate that our approach outperforms the state-of-the-art works.

INDEX TERMS Deep learning, intelligent transportation system, RDBDGN, traffic flow prediction.

I. INTRODUCTION

In contemporary society, as the traffic conflict is increasingly prominent, accurate and effective traffic flow prediction is becoming more and more important for improving road traffic efficiency and easing traffic congestion. The elevated highway, which alleviates the pressure caused by the excessive concentration of urban traffic to a great extent, is a essential part of urban traffic. Therefore, the realization of accurate flow prediction for elevated highways is of great help to address the issue of urban traffic congestion, and to provide guiding advice for individual travelers or public traffic planning. Traffic flow prediction refers to the prediction of future traffic flow information based on historical traffic flow data. In practice, traffic flow prediction can be divided into short-term (5-30 min) and long-term (over an hour) according to the length of projection time [1]. Compared with the short-term, long-term traffic flow prediction (especially one day prediction) is more favorable for individual travelers to make early decisions for the next day's travel. However, with the increase of prediction interval, the correlation between traffic flow data decreases and the randomness increases, which brings more difficulties to the long-term traffic flow prediction. By now, a considerable number of algorithms have been proposed for short-term traffic flow prediction. In general, existing traffic flow prediction methods can be classified into 3 major categories: traditional statistical methods, machine learning based models and deep learning based approaches.

In terms of the traditional statistical methods, most methods tried to build data-driven statistical models based on historical traffic data. For instance, an analytical method named autoregressive integrated moving-average (ARIMA) is widely used in representation of time-series data, and it was first used to predict short-term traffic flow in [2]. Based on Space-Time Autoregressive Integrated Moving Average (STARIMA), Duan et al. proposed a extended STARIMA model with time-varying lags for short-term traffic flow estimation [3]. Besides, other statistical models such as Kalman filtering based model [4] and stochastic Lagrangian traffic flow model [5] were also applied in traffic prediction.

Nevertheless, the traditional statistical methods mentioned above belong to linear models, which can hardly reflect the stochastic and nonlinear nature of traffic flow data. Afterwards, a great deal of machine learning methods including

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

K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Support Vector Regression (SVR) have been presented for traffic flow prediction as well. Sun *et al.* focused on flow-aware parameter and proposed weighted parameter tuples KNN for traffic prediction [6]. In [7], Feng *et al.* proposed an adaptive multi-kernel SVM algorithm to predict short-term traffic flow. Based on SVR model, many improved models such as Seasonal SVR [8] and Online-SVR [9] were employed to improve the forecasting accuracy. Moreover, an Artificial Neural Networks (ANN) model was implemented for short-term traffic flow prediction [10]. In [11], Xu *et al.* developed a spatiotemporal Bayesian multivariate adaptive-regression splines (ST-BMARS) model to investigate short-term freeway traffic flow prediction. Oh *et al.* combined Gaussian mixture model with an artificial neural network to form a multifactor pattern recognition model for urban traffic flow prediction [12]. Zhao and Sun developed a fourth-order Gaussian process dynamical model (GPDM) with the weighted KNN embedded to estimate future traffic flow [13]. In [14], a time series extreme learning machine model was also built for dealing with traffic prediction problems. Later, a kernel extreme learning machine (KELM) method was implemented to forecast short-term traffic flow [15].

There is no denying that machine learning methods have achieved better results in traffic flow prediction to some extent, yet, most of them use shallow models that have many limitations. Since deep neural networks have deeper and more complex structures, and have excellent ability to exploit the characteristics of traffic data, a varying number of traffic flow prediction methods based on deep learning have been proposed in recent years. Koesdwiady *et al.* [16] incorporated deep belief networks (DBNs) and data fusion to analyse the correlation between weather conditions and traffic flow, and realized the traffic flow prediction. DBNs were also adopted to predict short-term traffic flow based on big data in [17] and [18] successively. In addition, Stacked Auto Encoder(SAE) model has been used to predict the traffic flow in [19]. And then, a novel model based on SAE named stacked autoencoder Levenberg–Marquardt model (SAE-LM) was presented for traffic flow forecasting [20]. Arif *et al.* employed deep learning and parametric regression as the traffic flow prediction model [21]. In [22], a deep learning neural network DNN was built to estimate traffic conditions using big data. Liu et al. combined convolution and Long Short-Term Memory (LSTM) to generate a Conv-LSTM module to extract the spatial-temporal characteristics of the traffic flow [23]. A cascaded artificial neural network (CANN) was first applied as a novel depth model to forecast traffic flow in [24]. Zheng *et al.* proposed a hybrid model with BP, ε-SVR and LSTM for traffic flow prediction [25]. Recently, a deep spatio-temporal residual network (ST-ResNet) model has been applied for short-term urban traffic flow prediction both in [26] and [27].

Summarily, the existing studies have made some progress in short-term traffic prediction, but the problem of long-term prediction has not been well solved. Since Long-term traffic flow prediction is considered as a challenging problem in ITS, in this paper, considering the advantage of deep learning, especially Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN), we develop a residual deconvolution based deep generative network (RDBDGN) and apply it to deal with the problem of long-term traffic flow prediction on elevated highways. The flow information of the whole day in the future can be predicted more accurately by our model based on the historical traffic flow information.

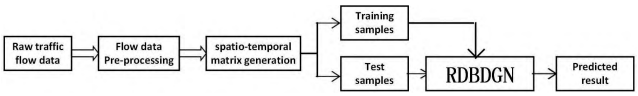


FIGURE 1. The process of our proposed RDBDGN model for traffic flow prediction.

Firstly, the original traffic flow data are transformed into spatio-temporal matrices, then these matrices are regenerated to three kinds of matrices with different time scales. Three flow matrices representing three different historical days form a group at each time scale. Therefore, the input to the RDBDGN model is three groups of matrices representing different time scales and containing flow information of three different historical days. The proposed RDBDGN model is mainly composed of a generator and a discriminator. In the generator module, we develop a personalized design model which combines residual net and deconvolutional neural network (RDNN) to learn and extract the multiscale spatio-temporal features of the traffic flow at three time scales. In the discriminator module, we employ a multi-channel CNN to distinguish the results of the generator from the real values, then, by optimizing the adversarial training process, more accurate results can be obtained. Experimental results have proved that our model is effective and it outperforms the state-of-the-art works for long-term traffic flow prediction.

II. PROPOSED METHODOLOGY

Figure 1 shows the process of our proposed approach, this paper aims at realizing a long-term traffic flow prediction for a day. The original data are collected by detectors deployed on the elevated highways at certain time intervals, and there is a certain position interval between these detectors. These data are first pre-processed and cleaned to remove abnormal data, and then transformed into spatial-temporal matrices, each matrix can be viewed as an image that fuses one-day spatial and temporal information of traffic flow. Finally, we randomly select most samples from these matrices as the training samples for the RDBDGN model, and the rest are used as the test samples to evaluate the predicted flow value of the next day.

A. TRAFFIC FLOW MATRIX GENERATION

Traffic flow data are collected by the detectors deployed in specific areas of the road at a certain time interval. At each time point, the value of each detector depends on the value of its adjacent upstream and downstream detectors.

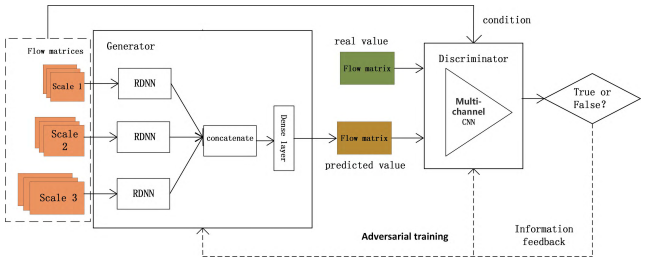


FIGURE 2. The overall structure of the proposed RDBDGN model.

Therefore, there is a spatio-temporal correlation between the traffic flow data. Usually, traffic flow data is transformed into a one-dimensional vector as the input of the model in most traffic flow prediction, which will ignore the spatio-temporal correlation of the traffic flow data. Since the spatial position and temporal variation are equally important information for traffic flow, taking into account the spatial and temporal correlations of traffic flow data inherently, we convert the original flow data with spatiotemporal complexity into a two-dimensional spatial-temporal matrix to retain the spatial and temporal information of the data. Let the x and y axes of the matrix represent the time dimension and the space dimension respectively, thus the mathematical form of the spatio-temporal flow matrix can be expressed as:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

The matrix X represents the traffic flow information for a day. Where m and n are the number of loop detectors and the number of time intervals respectively, x_{ij} represents the corresponding traffic flow value at position i and time j, and the unit for x_{ij} is vehicles per hour (veh/h). Then, we normalize the data using the maximum minimum value normalization method which is defined as:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

where x_{norm} indicates the normalized data, x is the original data, x_{max} and x_{min} are the maximum and minimum values of the original data respectively.

B. RESIDUAL DECONVOLUTION BASED GENERATIVE NETWORK

In order to adapt to the transportation environment, and build a more robust model to improve the accuracy of long-time traffic flow prediction, we propose the personalized design model RDBDGN. Figure 2 illustrates the overall structure of our proposed model. The whole model can be regarded as a complex Generative Adversarial Network (GAN), which is comprised of two adversarial modules: generator and discriminator. The input to the generator has three groups of flow matrices corresponding to three different time scales, and each group of matrices contains three matrices representing the flow information of different historical days. Meanwhile, we take the multi-scale flow matrices as condition ($\hat{C}$) input to the discriminator. Besides, the predicted result $G(\hat{C})$ yielded by the generator and a flow matrix representing the forecast day (real value) as a training label are fed to the discriminator as well. It has been proved that the performance of GAN can be improved when the generator was not conditioned on noise [28], therefore, the generator of our model only employs the condition $\hat{C}$ as the input.

In our work, the generator is mainly a combination of 3 residual deconvolutional neural networks (RDNNs) with the same structure, which can be regarded as 3 subnets. These subnets are employed for extracting both the spatial and temporal features. After that, the feature maps generated by the 3 subnets are concatenated and then directly delivered to a dense layer to generate the predictive value. Our discriminator is a multi-channel CNN module, which aims at judging whether the results generated by the generator are true or false, and feeding information back to the generator continuously so as to improve the prediction accuracy of the model.

1) THE INPUT FLOW MATRIX

To better capture the spatial and temporal distribution of traffic flow data from multiple time scales, and learn more robust input characteristics, 3 groups of matrices with time intervals of 10, 20 and 30 minutes respectively, are generated by resampling original data which has an interval of 5 minutes. The original flow data may be noisy, these matrices with three time scales enrich the spatiotemporal characteristics of traffic flow and reflect better traffic congestion patterns, therefore, they are chosen as the inputs of the generator. Moreover, since the flow data has obvious periodicity and temporal correlation, for each time scale, the input flow matrices incorporate three different historical days: the previous day, the previous week on the same day, and the previous two weeks on the same day are taken into account. Namely, to predict the flow of the $(T+1)th$ day, we need the historical flow information of the Tth day, $(T-6)th$ and $(T-13)th$ day. Thus in each group of matrices with the same time scale S_i (i $= 1, 2, 3$), the three input flow matrices representing different historical days can be expressed as follows:

$$X(T, S_i) = \begin{bmatrix} x_{11}(T, S_i) & \cdots & x_{1n}(T, S_i) \\ x_{21}(T, S_i) & \cdots & x_{2n}(T, S_i) \\ \cdots & \cdots & \cdots \\ x_{m1}(T, S_i) & \cdots & x_{mn}(T, S_i) \end{bmatrix} \tag{3}$$

$$X(T-6, S_i) = \begin{bmatrix} x_{11}(T-6, S_i) & \cdots & x_{1n}(T-6, S_i) \\ x_{21}(T-6, S_i) & \cdots & x_{2n}(T-6, S_i) \\ \cdots & \cdots & \cdots \\ x_{m1}(T-6, S_i) & \cdots & x_{mn}(T-6, S_i) \end{bmatrix} \tag{4}$$

$$X(T-13, S_i) = \begin{bmatrix} x_{11}(T-13, S_i) & \cdots & x_{1n}(T-13, S_i) \\ x_{21}(T-13, S_i) & \cdots & x_{2n}(T-13, S_i) \\ \cdots & \cdots & \cdots \\ x_{m1}(T-13, S_i) & \cdots & x_{mn}(T-13, S_i) \end{bmatrix} \tag{5}$$

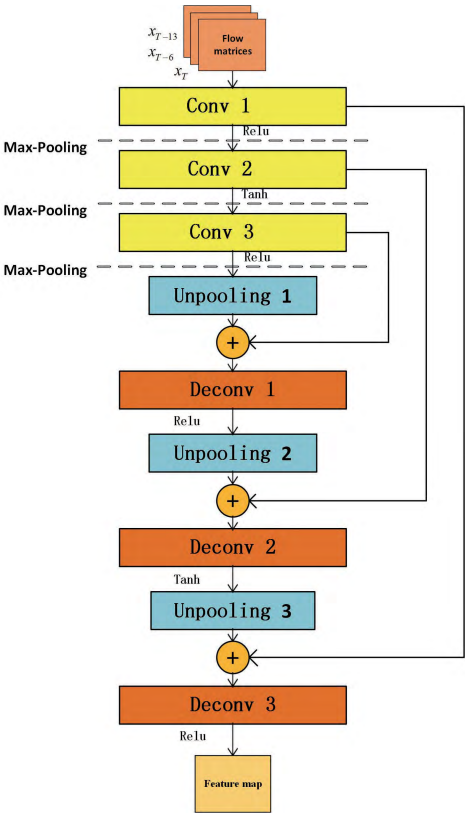


FIGURE 3. The structure of the RDNN model, where Conv represents the convolution layer, Max-Pooling represents the max pooling layer, unpooling denotes the unpooling layer and Deconv denotes the deconvolution layer.

where $X(T, S_i)$, $X(T-6, S_i)$ and $X(T-13, S_i)$ respectively represent the flow information matrices with a fixed time scale S_i of the Tth, $(T-6)th$ and $(T-13)th$ days. m and n also denote the number of loop detectors and the number of time series respectively, and the values of n are different at different time scales.

2) THE RESIDUAL DECONVOLUTION BASED GENERATOR

It is commonly accepted that GAN works by adversarial training between the generative network and discriminator network. In our model, for the generator, we develop the personalized design model (RDNN) based on CNN and apply it to learn and capture the spatio-temporal features implicit in the flow data for each time scale.

Figure 3 shows the structure of the RDNN model at one time scale. Our RDNN model incorporates two parts, the first part is the residual net module and the second part is the deconvolutional neural network module. First, we take advantage of the excellent performance of the residual net in deep neural network by adding residual units to extract the features of traffic flow. Then, we employ the deconvolutional network module to restore the information lost in the process of feature extraction and ensure the consistency of the dimension sizes of the two matrices for the residual operation. After a series of feature extraction operations from the lowest level to the highest level, finally, through a deconvolution layer, the advanced abstract feature map of this subnet is generated according to extracted comprehensive spatial and temporal features of flow data at a corresponding time scale.

As shown in figure 3, the RDNN has 12 layers, including a series of convolution, pooling, unpooling and deconvolution layers.

The convolutional layer is the most critical part for the model to learn the complex spatial and temporal characteristics of traffic flow data. In the convolution layer, firstly, the input image or feature map of previous layer is convolved by multiple trainable kernels to extract the important spatial and temporal features and generate the feature maps of this layer. Then the bias is added and the activation function is used to nonlinearize the convolutional result to yeild more complex characteristics. Assuming the number of feature maps of the previous layer is c^{l-1}, the output of the convolutional layer can be expressed as:

$$x_j^l = f\left(\sum_{i=1}^{c^{l-1}} x_i^{l-1} * k_{ij}^l + b_j^l\right) \tag{6}$$

where l indicates the layer index and j indicates the index of feature map at the lth layer, x_i^{l-1} represents a input feature map of the $(l-1)$ layer, $*$ indicates the convolution operation, x_j^l, k_{ij}^l, b_j^l represent the output feature map, kernel weights and bias at the lth layer respectively, f is the activation function. In our model, after the first, second and third convolution layers, we adopt the rectified liner unit active function, the *tanh* active function, and the rectified liner unit active function respectively. The rectified liner unit active function and the *tanh* active function are defined as (7) and (8):

$$f(x) = max(0, x) \tag{7}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{8}$$

The pooling layer is connected to the convolutional layer to filter the redundant information of the characteristics of the traffic flow and retain the significant spatial-temporal characteristics information. Therefore, the function of the pooling layer is to abstract the feature signals in a certain neighborhood of the feature map in the convolutional layer, which can greatly reduce the training parameters. In addition, the problem of overfitting can be reduced, and the invariance of features can be obtained. In this paper, we adopt the max pooling to down sample the result after the convolutional layer.

The deconvolution network can be viewed as the inverse process of the convolution network, which mainly consists of the deconvolution layer and the unpooling layer. For the deconvolution layer, since the backward propagation of the convolutional layer is the forward propagation of the deconvolution layer, we employ a method called transpose convolution to realize the deconvolution operation. Namely, to take the transpose of the same convolution kernel as the deconvolution kernel, and convolve the input with it. Whereupon,

the output of the deconvolution layer can be expressed as:

$$x_j^l = f\left(\sum_{i=1}^{c^{l-1}} x_i^{l-1} * (k_{ij}^l)^R + b_j^l\right) \tag{9}$$

where, $*$ also denotes the convolution operation, R indicates the transpose of the matrix, and f is the activation function. Here, the activation functions after the first, second and third deconvolution layers are the same as the activation functions after the first, second and third convolution layers.

Unpooling is the inverse operation of pooling, and it is impossible to restore all the original information from the results generated by pooling operation. Therefore, the process of pooling only retains the main information, and some information is omitted. Since pooling is an irreversible process, we need to record the coordinate position of the maximum value during the pooling operation, then, this recorded position is filled with the maximum value, and the other positions are filled with 0 in the process of unpooling. Therefore, the output feature map j at the *lth* unpooling layer can be written as:

$$x_j^l = unmp(x_j^{l-1}, argmax_j) \tag{10}$$

where, *unmp* denotes the unpooling operation, x_j^{l-1} is a feature map of previous pooling layer and $argmax_j$ indicates the index of the position where the maximum value is.

After obtaining the feature maps with the same output sizes of 3 time scales generated by the 3 subnets, we combine them together and deliver the fused information to a fully connected layer to decode the predicted results. In the fully connected layer, let x^i represent the *ith* input matrix of the full connection layer with a dimension of $r \times c$, it is then reshaped as a vector $\mathbf{v}^i$:

$$\mathbf{v}^i = \left[x_{11}^i, \ldots, x_{1c}^i; x_{21}^i, \ldots, x_{2c}^i; \ldots; x_{r1}^i, \ldots, x_{rc}^i\right] \tag{11}$$

Assuming that the input to the full connection layer has n matrices, they all can be reshaped and concatenated to form a longer vector which can be represented as:

$$\mathbf{V} = [\mathbf{v}^1; \mathbf{v}^2; \ldots; \mathbf{v}^i; \ldots \mathbf{v}^n] \tag{12}$$

Since the final prediction is a flow matrix which has the same size as the input matrix with 10-minute intervals, we continue to reshape the output vector of the full connection layer into n matrices. It means that the number of nodes in the output layer depends on the size of corresponding input matrix of 10-minute intervals. All neurons in the fully connected layer and output layer are globally connected, the output of the *kth* predicted flow node F_k for the next day can be written as:

$$F_k = f(\sum_{i=1}^{r \times c \times n} w_i y_i + b_i) \tag{13}$$

where $y_i \epsilon \mathbf{V}$, which denotes one of the nodes in the fully connected layer, w_i and b_i are the corresponding weight and bias, respectively. f is the sigmoid activation defined as follows:

$$f(x) = \frac{1}{1 + e^x} \tag{14}$$

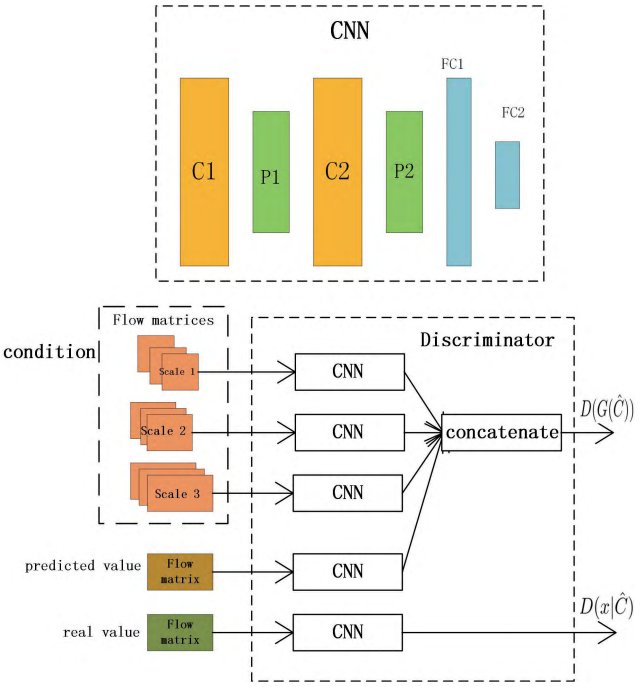


FIGURE 4. The structure of the discriminator, C indicates the convolution layer, P indicates the pooling layer and FC represents the fully connected layer.

3) THE DISCRIMINATOR AND THE CNN MODEL

In this paper, we make full use of remarkable learning ability of CNN to build the discriminator, which can be regarded as a binary classifier, and is used to distinguish the predicted value generated by the generator from the real value as accurately as possible. Furthermore, to make a more reliable decision about the input and output of the generator, we condition the discriminator on the multi-scale flow information which are fed to generator. As shown in figure 4, our discriminator is composed of multi-channel CNN, and the multi-scale flow matrices are taken as the condition. The CNN module in the discriminator contains 2 convolution layers, 2 max pooling layers and 2 fully connected layers. The first fully connected layer adopts the rectified liner unit active function, and the activation function of the output layer is sigmoid. The output contains two parts: $D(G(\hat{C}))$ and $D(x|\hat{C})$, which denote the probability that the output generated by the generator came from training sample, and the probability that the real value came from training sample respectively.

4) MODEL OPTIMIZATION

To train the parameters, a loss function is needed to measure the prediction accuracy of the model. During the training phase, the generative network (G) and the discriminator network (D) optimize the output as they compete with each

other; the generator makes attempts to maximize the probability that its output is true judged by the discriminator, while the discriminator tries to minimize the probability that the generator output is true. G and D are trained alternately. In each step of the training, one of them is fixed, while the other performs the optimization of parameters. As a whole, D and G play a two-player minimax game, and the whole objective function to be optimized is defined as follows:

$$\min_G \max_D V(D,G) = E_{x\sim P(x)}[log(D(x|\hat{C})] + E[log(1-D(G(\hat{C})))] \quad (15)$$

where E represents the empirical estimation of the expected value of the probability, $x \sim P(x)$ means the training data x comes from the real distribution $P(x)$. $\hat{C}$ denotes the given condition, which refers to the three scales flow information. $G(\hat{C})$ denotes the output generated by the generator. $D(x|\hat{C})$ indicates the probability that x came from training data and the discriminator is conditioned on $\hat{C}$. To conclude, D works for maximizing $log(D(x|\hat{C}) + log(1-D(G(\hat{C})))$ and G works for minimizing $log(1-D(G(\hat{C})))$.

The goal of generator is to fit the distribution of training samples and deceive the discriminator as far as possible. In our model, the overall loss function of G is defined as:

$$L_G = \alpha L_{fake\_to\_real} + L_{mse}(F_k{}', F_k) + L_{reg} \quad (16)$$

where, α is the factor that controls the adversarial degree between the G and the D, and here we set it to 0.01. $L_{fake\_to\_real}$ is the loss that denotes the predicted result (false sample) is true, which aims to make the predicted value generated by the G more approximate to the real value and is defined as:

$$L_{fake\_to\_real} = -log(D(G(\hat{C}))) \quad (17)$$

L_{mse} is mean squared error. Assuming there are N training samples, $F_k{}'$ and F_k represent the real flow value and predicted flow value, then the L_{mse} can be represented as:

$$L_{mse} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{F_k}' - \mathbf{F_k})^2 \quad (18)$$

To predict traffic flow, we need to use L_{mse} to calculate the difference between the ground truth and the predicted result, and our goal is to minimize the difference as far as possible, thus L_{mse} is the most essential part of the loss function in generator.

L_{reg} is the L2 norm (weight decay), which is defined as:

$$||v||_2 = \sqrt{\sum_{i=1}^{k}|v_i|^2} \quad (19)$$

where $||v||_2$ denotes the square root of the sum of the squares of k vector elements v_i (i = 1, 2 . . . k). We add L_{reg} to the loss function in order to avoid the problem of overfitting.

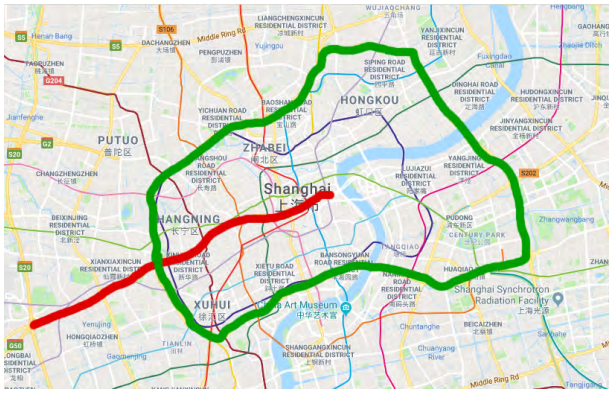


FIGURE 5. Marking of two elevated highways. Red and green bold lines respectively mark Yan'an elevated highway and Neihuan elevated highway.

The goal of discriminator is to judge whether the sample is from real distribution. Thus, here the loss function of D is defined as:

$$L_D = L_{fake} + L_{real} \quad (20)$$

where L_{fake} is the loss that the predicted value is false, which defined as:

$$L_{fake} = -log(1-D(G(\hat{C}))) \quad (21)$$

L_{real} is the loss that the real value is true, which defined as:

$$L_{real} = -log(D(x|\hat{C})) \quad (22)$$

During the training phase, G and D play against with each other until they reach the Nash equilibrium. At this time, the sample generated by G is almost indistinguishable from the real sample, and the probability determined by D infinitely approaches to 1/2, the optimal solution is obtained, as the D finds it is hard to distinguish fake samples. In this paper, the loss function of both G and D are optimized by Adam method.

III. EXPERIMENTAL RESULTS

A. EXPERIMENTAL DATA AND SETTING

The factual flow data with a sampling period of 5 min are collected from detectors deployed on Yan'an and Neihuan elevated highways of year 2011. As two major roads of urban traffic network in Shanghai, China, these two elevated highways have expanded the city's transportation capacity to a large extent. To collect traffic flow data, there exists a loop detector every 400 meters on each elevated highway, and the number of detectors deployed on the Yan'an and Neihuan elevated highways are 35 and 72, respectively. As shown in figure 5, Yan'an and Neihuan elevated highways are respectively marked in red and green.

As described in part A of section II, we need to transform the original data into spatio-temporal flow matrices for each day. Nevertheless, the original matrices may contain some errors caused by detector failure or external factors, so they need to be pre-processed to remove the abnormal

elements first. To ensure that the flow value in the reasonable value range, values greater than 1000 or or equal to 0 are replaced by the average of adjacent values. Considering there is almost no traffic at night and it is meaningless to make prediction, we choose data collected from 7 am to 10 pm for experiments. After pre-processing, in order to exploit the rich spatial-temporal characteristics of flow parameters from different time scales, the flow matrices with time intervals of 5 minutes for each day is resampled to generate 3 kinds of matrices with time intervals of 10, 20 and 30 minutes. Finally, all the matrices for the year of 2011 are gathered to generate a dataset for each elevated highway.

For Yan'an elevated highway, owing to missing data from March 20 to March 23, only 361 days of data are available for the experiment actually. As we need to use the previous two weeks' historical data as input, thus the number of samples of Yan'an and Neihuan are 347 and 351 respectively, which are equal to the total number of days minus 14. To evaluate the performance of our proposed model, we randomly select 30 samples as the test set for each dataset, and the rest of the samples are considered as the training set.

As mentioned above, the three groups of the input matrix correspond to three time scales with time intervals of 10, 20 and 30 minutes respectively. In regards to the size of three groups of input matrix, in terms of the time dimension, since we choose data collected from 7 am to 10 pm, then for 10-minute sampling interval, 20-minute sampling interval, and 30-minute sampling interval, there are 90, 45 and 30 time series respectively. Therefore, the widths of three groups of input matrix are 90, 45 and 30. In terms of the space dimension, we map the spatial sequence of the detectors directly to the space dimension, then the height of the three matrices is equal to the number of detectors of corresponding elevated highway. As a result, for Yan'an elevated highway, the resolutions of three groups of input matrix are 35×90, 35×45, and 35×30. For Neihuan elevated highway, the resolutions of three groups of input matrix are 72×90, 72×45, and 72×30 respectively.

In our experiment, the inputs to the generator are three four-dimensional matrices corresponding to three time scales, and the four dimensions refer to batch size, the number of detectors, the number of time series, and the number of historical days. Batch size is set to 10 in the process of network training. Consequently, for Yan'an elevated highway, the dimension size of three inputs to the generator are (10, 35, 90, 3), (10, 35, 45, 3) and (10, 35, 30, 3). The output of the generator is a matrix with 10-minute intervals, thus both the output of generator and the label which are fed to discriminator have the same dimension size with (10, 35, 90, 1), while the two outputs of the discriminator are scalars. For Neihuan elevated highway, the dimension size of three inputs to the generator are (10, 72, 90, 3), (10, 72, 45, 3) and (10, 72, 30, 3). Similarly, The dimension size of two inputs to the discriminator are (10, 72, 90, 1), and the two outputs of the discriminator are scalars as well.

TABLE 1. Parameter configuration of residual deconvolution module (RDNN) in generator.

Layers	Name	Description
Layer1	Convolution 1	64 kernels with $5 \times 5 \times 3$ size
Layer2	Max pooling 1	Kernel size = 2×2, stride = 2
Layer3	Convolution 2	64 kernels with $3 \times 3 \times 64$ size
Layer4	Max pooling 2	Kernel size = 2×2, stride = 2
Layer5	Convolution 3	128 kernels with $3 \times 3 \times 64$ size
Layer6	Max pooling 3	Kernel size = 2×2, stride = 2
Layer7	Unpooling 1	Kernel size = 2×2, stride = 2, output shape=output shape of Convolution 3
Layer8	Deconvolution 1	Kernel size = 3×3, output shape=output shape of Max-pooling 2
Layer9	Unpooling 2	Kernel size = 2×2, stride = 2, output shape=output shape of Convolution 2
Layer10	Deconvolution 2	Kernel size = 3×3, output shape=output shape of Max-pooling 1
Layer11	Unpooling 3	Kernel size = 2×2, stride = 2, output shape=output shape of Convolution 1
Layer12	Deconvolution 3	Kernel size = 5×5, output shape=output shape of input

TABLE 2. Parameter configuration of CNN module in discriminator.

Layers	Name	Description
Layer1	Convolution 1	32 kernels with $5 \times 5 \times 1$ size
Layer2	Max pooling 1	Kernel size = 2×2, stride = 2
Layer3	Convolution 2	64 kernels with $3 \times 3 \times 32$ size
Layer4	Max pooling 2	Kernel size = 2×2, stride = 2
Layer5	FC 1	128 neuron nodes
Layer6	FC 2	1 neuron node

The hardware environment in which we conduct our experiments includes the server with i7-5820K CPU, 48GB memory and NVIDIA GeForce GTX1080 GPU, and we adopt the TensorFlow framework of deep learning to implement our model. The parameter configuration of our proposed model are shown in tables 1 - 2. In this paper, parameters and network structure for 2 elevated highways are identical. The strides for all the convolution kernels are set to 1×1. The learning rates of both generator and discriminator are set as 0.0001, and the total number of our network iterations is 30000.

Note that the parameter settings for the three RDNN subnets are the same.

B. RESULTS AND ANALYSIS

In order to visually illustrate the predictive performance of our model from the visualized results, we visualize the flow matrix as a heat map, which can visually reveal to us the real information of traffic flow in a day. In a heat map, the x-axis denotes time series of a day. There are 90 time series since the time range is from 7am to 10pm, and the time interval is 10 minutes here. The y-axis indicates the number of the detectors, which also reflects the position of detectors. Different shades of color in the heat map are mapped to flow values, the darker the color is the smaller the flow value is. Figure 6 shows the heat maps of Yan'an

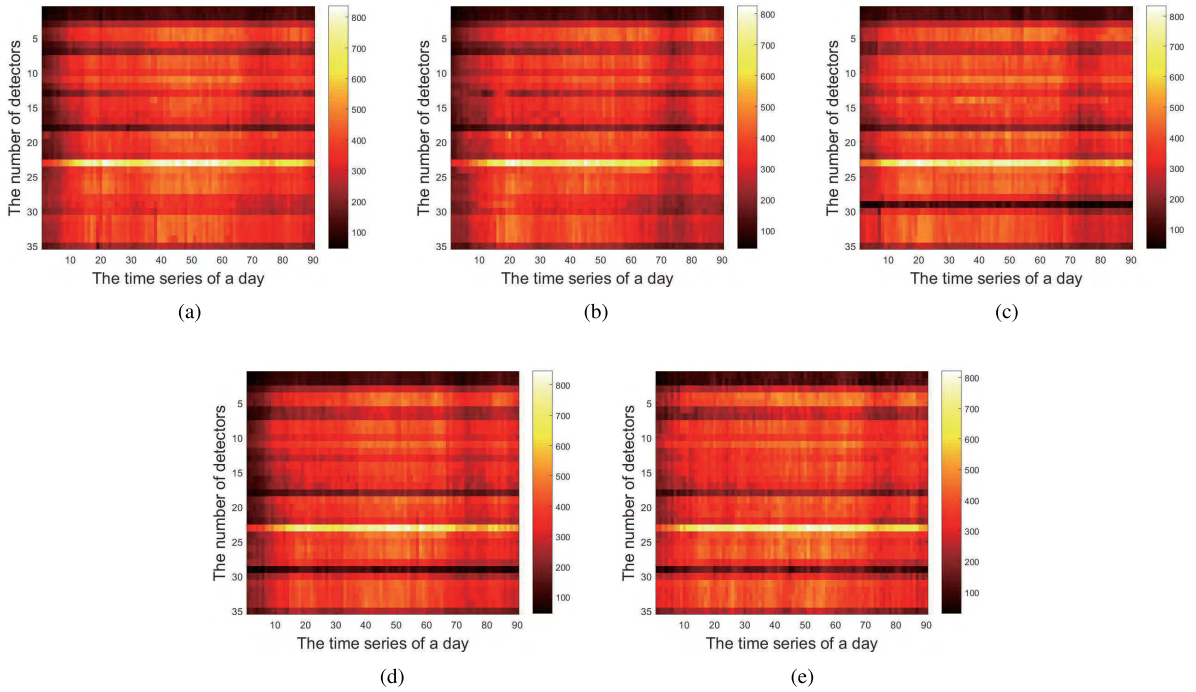


FIGURE 6. Visualized heat maps of flow matrices of Yan'an elevated highway. (a) Visualized heat map of the 145th day. (b) Visualized heat map of the 152nd day. (c) Visualized heat map of the 158th day. (d) Ground truth of the 159th day. (e) Predicted result of the 159th day.

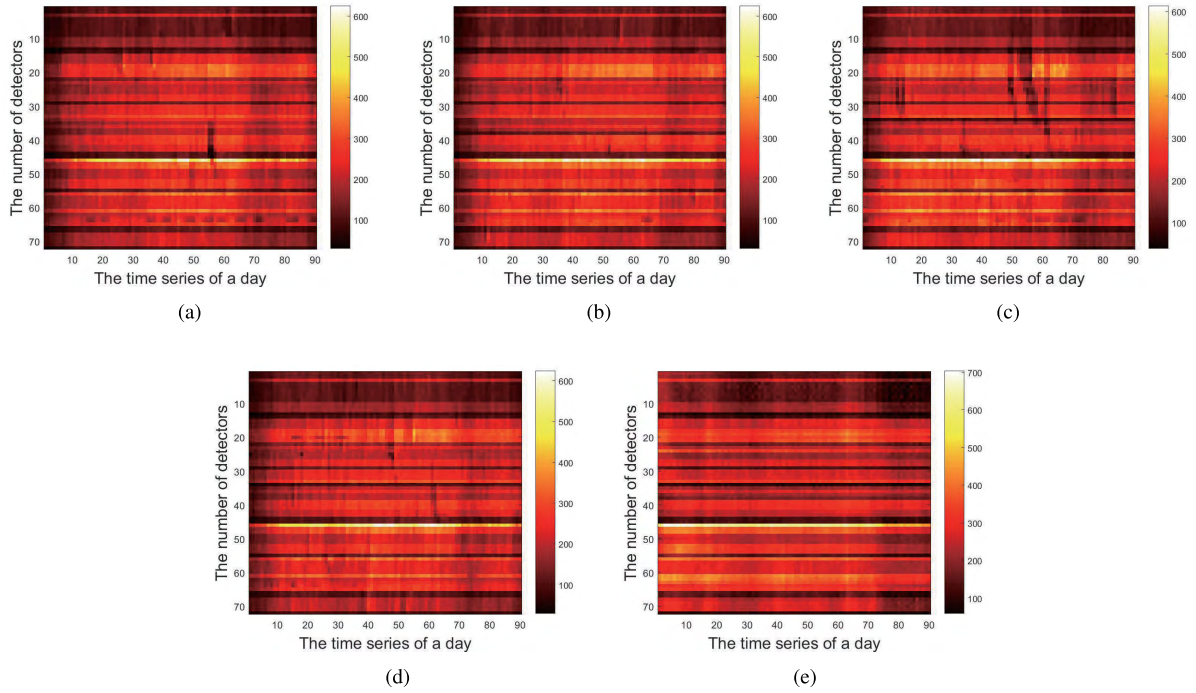
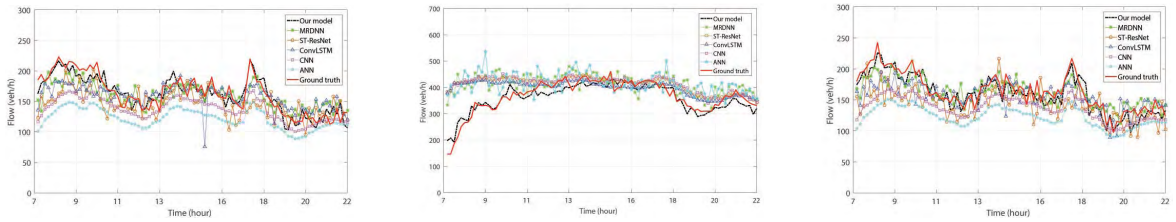


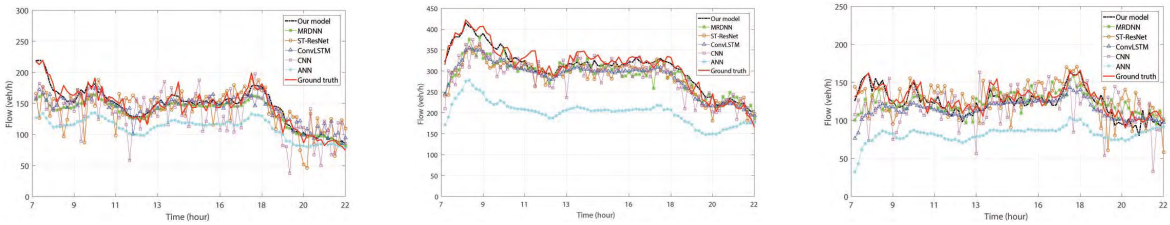
FIGURE 7. Visualized heat maps of flow matrices of Neihuan elevated highway. (a) Visualized heat map of the 170th day. (b) Visualized heat map of the 177th day. (c) Visualized heat map of the 183th day. (d) Ground truth of the 184th day. (e) Predicted result of the 184th day.

elevated highway, in which the first line includes the heat maps of the 145th, 152nd and 158th days respectively from left to right; and the second line contains the heat maps of the ground truth of the 159th day and its corresponding predicted result respectively from left to right. In this case, we need to predict the flow information for the 159th day, as the flow data of 158th day has a temporal correlation with the flow data of 159th day, while the flow data of 152nd day and the 145th day have the periodic correlation with the flow data of 159th day, thus the historical flow information of 158th,



(a) (b) (c)

FIGURE 8. Curves of predicted values and the corresponding ground truth of Yan'an elevated highway. (a) Prediction and ground truth of the 1st loop detector of the 182nd day. (b) Prediction and ground truth of the 8th loop detector of the 355th day. (c) Prediction and ground truth of the 2nd loop detector of the 175th day.



(a) (b) (c)

FIGURE 9. Curves of predicted values and the corresponding ground truth of Neihuan elevated highway. (a) Prediction and ground truth of the 7th loop detector of the 151st day. (b) Prediction and ground truth of the 53th loop detector of the 314th day. (c) Prediction and ground truth of the 14th loop detector of the 137th day.

152nd and 145th days are needed. Figure7 shows the heat maps of Neihuan elevated highway, in which the heat maps of the 170th, 177thnd and 183th days are represented in the first row; and heat maps of the ground truth of the 184th day and its corresponding predicted result are represented in the second row. As shown in these two figures, some sub-figures in them look very similar such as (b) and (d), (c) and (d) in figure 6, which reflect the periodicity and temporal correlation of the traffic flow. It can also be seen visually from the heat maps of ground truth and predicted result that the predicted results of our model is quite close to the ground truth.

Figures 8-9 show the curves of predicted values and the corresponding ground truth of Yan'an elevated highway and Neihuan elevated highway respectively. In each figure, subgraphs from left to right represent flow values of different detectors on different days. In these figures, the red solid curves represent the ground truth, and the black broken curves represent the predicted values of our model, other curves with different colors and types represent the results of compared methods. It can be seen from these figures that the predicted results generated by the RDBDGN model are most approximate to the ground truth.

In this paper, Mean Relative Error (MRE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are used to measure the prediction performance. MAE and RMSE can assess the absolute error between the prediction and the reality, and MRE can evaluate the relative error between them. The formulas of MSE, MRE and MAE are defined as follows:

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{y}' - \mathbf{y}|}{\mathbf{y}} \tag{23}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{y}' - \mathbf{y}| \tag{24}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}' - \mathbf{y})^2} \tag{25}$$

where $\mathbf{y}'$ is the predicted value, $\mathbf{y}$ is the reality and N represents the number of samples in test set.

For further prove the effectiveness of the proposed model, we compare our presented method with the methods of classical ANN, CNN, and the state-of-the-art approach: ST-ResNet [27]. Since the Convolutional LSTM (ConvLSTM) has the advantage of learning spatial and temporal characteristics simultaneously, we have made a comparison with it as well. What's more, two variants of our model: *RDBDGN_V* and MRDNN, are also compared. *RDBDGN_V* has the same structure as our model except that there is no condition input to its discriminator. MRDNN is the multiscale residual deconvolutional neural network, which can be regarded as the multiscale RDNN without the GAN module.

In the comparative experiments, the number of neuron nodes in the hidden layer of ANN is 1500. CNN consists of 3 convolution layers, 2 max pooling layers and 2 fully connected layers. The ST-ResNet is composed of 3 components corresponding to 3 different time scales, each component is a residual network, then, these 3 residual networks are followed by a fully connected layer. The structure and parameter settings of the variant model are the same as the RDNN module in our model. The ConvLSTM contains 1 ConvLSTMCell and is unrolled by time step. Note that the input of ConvLSTM here are the same 3 historical days flow data as ours, thus the number of time step in ConvLSTM

TABLE 3. Comparison results for Yan'an elevated highway.

Models	MRE (%)	MAE	RMSE
RDBDGN	**11.25**	**32.78675**	**44.15930**
RDBDGN_V	11.671	34.22869	45.85653
ST-ResNet	15.6	49.21540	62.45258
MRDNN	14.554	43.01504	57.63739
ConvLSTM	14.498	46.44383	60.77586
CNN	16.64	49.55268	63.87733
ANN	21.152	59.41958	75.78244

TABLE 4. Comparison results for Neihuan elevated highway.

Models	MRE (%)	MAE	RMSE
RDBDGN	**11.140**	**21.53695**	**30.84762**
RDBDGN_V	11.388	22.58749	31.60150
ST-ResNet	18.1	28.31851	40.36108
MRDNN	15.5	26.55174	37.07883
ConvLSTM	16.442	27.26390	37.89466
CNN	17.966	29.45661	40.4923
ANN	22.307	34.88531	45.66673

is 3. *RDBDGN_V* uses the same historical data as our model. Other comparison models including MRDNN all use the previous day's flow information to predict the value of the next day.

Finally, the comparison results are shown in Tables 3 - 4. As shown in these tables, the presented RDBDGN model has the lowest error and the best performance in comparsion with other 6 methods. Conventional machine learning approach ANN performs worst. CNN performs better than ANN due to its excellent ability to learn and extract the features of complex data automatically. For Yan'an elevated highway, ST-ResNet performs slightly better than CNN. In contrast, the variant (MRDNN) of ours performs much better due to the addition of the deconvolutional neural network module. It can be seen from the table 3 that although the MRE of ConvLSTM and MRDNN are very nearly the same, the MAE and RMSE of MRDNN are all lower than that of ConvLSTM. Obviously, our model gets the lowest error, as its MRE, MAE and RMSE are 11.25%, 32.78675 and 44.15930, respectively. As for Neihuan elevated highway, results present that ST-ResNet has barely improved in performance relative to CNN. However, the performance improvement of MRDNN is more obvious, and MRDNN works slightly better than ConvLSTM. Certainly, the MRE, MAE and RMSE metrics yielded by the our approach (RDBDGN), are the lowest values among all the listed methods, which are 11.140%, 21.53695 and 30.84762, respectively. As can be seen from table 3 and table 4, our model has a slight improvement in performance compared with the *RDBDGN_V*. In fact, the condition used by discriminator is not very binding on our model, while the ground truth is a strong constraint for our model, which constrains every pixel of the flow matrix in the process of training. Yet it is certain that the condition further constrains our model well, and the presented results have shown that the addition of condition in the discriminator contributes to the further improvement of our model performance.

To demonstrate the effectiveness of multi-scale fusion, we only take the historical flow matrices with a specific time scale as the input (the input data with single-scale), and

TABLE 5. Results correspond to different time scales for Yan'an elevated highway.

Models	MRE (%)	MAE	RMSE
RDBDGN	11.25	32.78675	44.15930
RDBDGN_V	11.671	34.22869	45.85653
Low_scale	14.151	42.03291	53.66273
Mid_scale	11.742	37.04004	47.92081
High_scale	14.056	44.20346	56.79403

TABLE 6. Results correspond to different time scales for Neihuan elevated highway.

Models	MRE (%)	MAE	RMSE
RDBDGN	11.140	21.53695	30.84762
RDBDGN_V	11.388	22.58749	31.60150
Low_scale	13.531	22.03384	29.96999
Mid_scale	16.528	31.90277	41.45170
High_scale	12.144	23.83852	32.46297

conduct experiments based on the *RDBDGN_V* model. The experimental results are shown in table 5 and table 6. In these two tables, *Low_scale*, *Mid_scale* and *High_scale* correspond to scale with 10-minute intervals, scale with 20-minute intervals and scale with 30-minute intervals respectively. It can be seen from these two tables that no specific time scale has the optimal result for all data sets. Since the data distribution is different at different time scales, we need to capture the spatial and temporal characteristics of traffic flow data from multiple time scales, therefore, multi-scale fusion is essential for the analysis on each time scale and it has been proved that the *RDBDGN_V* model performs much better when inputs are multi-scale fused data rather than single-scale data.

Last but not least, to further verify the validity of Eq.16 in this work, we have studied the different components of Eg.16. When α is set to zero, for Yan'an elevated highway, we obtained the results of MRE, MAE and RMSE as 13.078%, 36.65264 and 49.28449 respectively. For Neihuan elevated highway, we obtained the results of MRE, MAE and RMSE as 13.074%, 25.60196 and 34.65133 respectively. When L_{mse} is set to zero, the MRE, MAE and RMSE results of Yan'an elevated highway are 18.231%, 50.24229 and 65.07285 respectively, and the MRE, MAE and RMSE results for Neihuan are 17.844%, 34.26457, and 44.46853 respectively. These results suggest that when the weight factor α is set to zero, the discriminator does not work and there is no adversariality between the generator and the discriminator, which will affect the performance of the model. In comparison, when the loss L_{mse} is set to zero, the performance of the model decreases obviously. This is so because, for the problem of flow prediction, we train our model mainly by regressing to the ground truth. Therefore, the L_{mse} in the loss function of generator is crucial for training the model, which is why we add this term to the loss function of the generator.

Long-term traffic flow prediction heavily depends on spatio-temporal information and periodical information of historical traffic data. The proposed RDBDGN model takes into account multiscale spatial and temporal features, and strong temporal correlation features, which significantly contribute to making prediction. Moreover, we take full

advantage of the adversarial training of GAN, in the generator, we adopt residual idea and apply it to extract the spatial-temporal feature and periodic feature of traffic flow data, the problem of detail loss in feature extraction is solved simultaneously by employing deconvolution network module. In the discriminator, we first make full use of the excellent performance of CNN to optimize the adversarial training process of the network, which greatly contribute to enhancing the prediction accuracy. Then, we further improve the performance of our model by conditioning the discriminator on the multi-scale flow information. The above experimental results demonstrate that our approach obtains the best performance.

IV. CONCLUSION

In this paper, we take advantage of the outstanding performance of GAN to propose a novel residual deconvolution based deep generative network (RDBDGN), and apply it to improve the accuracy of long-term traffic flow prediction. First of all, raw traffic flow data are transformed into spatial-temporal matrices. Next, the proposed RDBDGN accepts 3 different time scales of traffic flow data, furthermore, considering the strong temporal correlation of traffic flow, flow information of 3 historical days are fused as the input at each time scale. The RDBDGN is essentially a complex GAN, which is comprised of a generator module and a discriminator module. The generator is mainly a combination of 3 identical residual deconvolutional neural networks (RDNN), which aims to fully extract the multiscale spatio-temporal and periodic features of the traffic flow and decode the flow of the next day. Then, the discriminator composed of multi-channel CNN module is added to make the predicted value more approximates to the real value by optimizing the adversarial training process so as to improve the accuracy of long-term traffic flow prediction. Two data sets of 2 main elevated highways in Shanghai, China are used to testify the efficiency and accuracy of our method. Finally, experimental results prove that the proposed model is robust and outperforms the state-of-the-art works.

REFERENCES

[1] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 99, pp. 144–163, 2019. doi: 10.1016/j.trc.2018.12.004.

[2] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using box-jenkins techniques," *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[3] P. Duan, G. Mao, C. Zhang, and S. Wang, "STARIMA-based traffic prediction with time-varying lags," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Nov. 2016, pp. 1610–1615.

[4] Z. H. Mir and F. Filali, "An adaptive Kalman filter based traffic prediction algorithm for urban road network," in *Proc. Int. Conf. Innov. Inf. Technol.*, 2017, pp. 1–6.

[5] K.-C. Chu, R. Saigal, and K. Saitou, "Stochastic Lagrangian traffic flow modeling and real-time traffic prediction," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, Aug. 2016, pp. 213–218.

[6] B. Sun, W. Cheng, P. Goswami, and G. Bai, "Flow-aware WPT k-nearest neighbours regression for short-term traffic prediction," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 48–53.

[7] 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 Trans. Intell. Transp. Syst.*, to be published.

[8] W.-C. Hong, "Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, nos. 12–13, pp. 2096–2107, 2011.

[9] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, 2009.

[10] R. More, A. Mugal, S. Rajgure, R. B. Adhao, and V. K. Pachghare, "Road traffic prediction and congestion control using artificial neural networks," in *Proc. Int. Conf. Comput., Anal. Secur. Trends*, 2017, pp. 52–57.

[11] Y. Xu, Q.-J. Kong, R. Klette, and Y. Liu, "Accurate and interpretable Bayesian MARS for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2457–2469, Dec. 2014.

[12] S.-D. Oh, Y.-J. Kim, and J.-S. Hong, "Urban traffic flow prediction system using a multifactor pattern recognition model," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2744–2755, Oct. 2015.

[13] J. Zhao and S. Sun, "High-order Gaussian process dynamical models for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2014–2019, Jul. 2016.

[14] Q. Zhang, D. Jian, R. Xu, W. Dai, and Y. Liu, "Integrating heterogeneous data sources for traffic flow prediction through extreme learning machine," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 4189–4194.

[15] Y. Xing, X.-J. Ban, and R. Liu, "A short-term traffic flow prediction method based on kernel extreme learning machine," in *Proc. IEEE Int. Conf. Big Data Smart Comput.*, Jan. 2018, pp. 533–536.

[16] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[17] R. Soua, A. Koesdwiady, and F. Karray, "Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory," in *Proc. Int. Joint Conf. Neural Netw.*, 2016, pp. 3195–3202.

[18] Y. Zhang and G. Huang, "Traffic flow prediction model based on deep belief network and genetic algorithm," *IET Intell. Transp. Syst.*, vol. 12, no. 6, pp. 533–541, 2018.

[19] 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.

[20] H.-F. Yang, T. S. Dillon, and Y.-P. P. Chen, "Optimized structure of the traffic flow forecasting model with a deep learning approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2371–2381, Oct. 2017.

[21] M. Arif, G. Wang, and S. Chen, "Deep learning with non-parametric regression model for traffic flow prediction," in *Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput., 16th Int. Conf. Pervasive Intell. Comput., 4th Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Aug. 2018, pp. 681–688.

[22] H. Yi, H. Jung, and S. Bae, "Deep neural networks for traffic flow prediction," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 328–331.

[23] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with Conv-LSTM," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.

[24] S. Zhang, Z. Kang, Z. Hong, Z. Zhang, C. Wang, and J. Li, "Traffic flow prediction based on cascaded artificial neural network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 7232–7235.

[25] Z. Zheng, L. Pan, and K. Pholsena, "Mode decomposition based hybrid model for traffic flow prediction," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2018, pp. 521–526.

[26] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 1655–1661.

[27] X. Wu, S. Ding, W. Chen, J. Wang, and P. C. Y. Chen, "Short-term urban traffic flow prediction using deep spatio-temporal residual networks," in *Proc. 13th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, May/Jun. 2018, pp. 1073–1078.

[28] D. Pathak, P. Krähenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

DI ZANG received the Ph.D. degree from Kiel University, Germany, in 2007. She was a Postdoctoral Researcher with the University of Minnesota, USA, in 2008. She is currently an Associate Professor with Tongji University, Shanghai, China. Her current research interests include deep learning, intelligent transportation systems, and computer vision.

YANG FANG received the bachelor's degree with Chongqing University, China. She is currently pursuing the master's degree with Tongji University, China. Her current research interests include deep learning and intelligent transportation systems.

ZHIHUA WEI received the B.S. and M.S. degrees from Tongji University, in 2000 and 2005, respectively, and the double Ph.D. degrees from Tongji University and Lyon2 University, in 2010, where she is currently a Professor. Her research interests include machine learning, image processing, and data mining.

KESHUANG TANG received the Ph.D. degree in transportation engineering from Nagoya University, in 2008. He was with The University of Tokyo as a Postdoctoral Research Fellow, and then at Tohoku University as a Project Assistant Professor. He is currently a Professor with the Department of Transportation Information and Control Engineering, Tongji University, China. His current research interests include driver behavior, signal control, and intelligent transportation systems.

JIUJUN CHENG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2006. In 2009, he was a Visiting Professor with Aalto University, Espoo, Finland. He is currently a Professor with Tongji University, Shanghai, China. His research interests include mobile computing and complex networks with a focus on mobile/Internet interworking and the Internet of Vehicles.