IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 August 2022, accepted 30 August 2022, date of publication 5 September 2022, date of current version 13 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204036

RESEARCH ARTICLE

GCN-GAN: Integrating Graph Convolutional Network and Generative Adversarial Network for Traffic Flow Prediction

HONGLING ZHENG[1], XIANG LI[2], YONGFENG LI[3], ZIQIN YAN[4], AND TINGHONG LI[5]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China
[2]The Fiftieth Research Institute of China Electronic Technology Group Corporation, Putuo, Shanghai 200331, China
[3]School of Cyberspace Security, Xi'an Jiaotong University, Xi'an 710000, China
[4]The Silk Road Infoport Company Ltd., Xi'an 710049, China
[5]China Mobile Communications Corporation Ltd., Beijing 100032, China

Corresponding author: Tinghong Li (charmsky@gmail.com)

This work was supported by the School of Information Science and Engineering, Lanzhou University.

ABSTRACT As a necessary component in intelligent transportation systems (ITS), traffic flow-based prediction can accurately estimate the traffic flow in a certain period and area in the future. However, despite the success of traditional research and current machine learning methods, traffic flow prediction models have limitations in terms of prediction accuracy and efficiency. In this work, we propose a novel traffic flow prediction model named Graph Convolution and Generative Adversative Neural Network (GCN-GAN), which leverages Graph Convolution Neural Network (GCN) module and Generative Adversative Neural Network (GAN) module to predict urban traffic flow. Firstly, the GCN module extracts historical traffic flow information in the graph structure. Secondly, the GAN module generates reliable traffic flow prediction results by adversative training. Additionally, GCN-GAN can parallelly generate prediction results rather than traditional one by one. Through experiments on the traffic flow dataset at multiple intersections, our GCN-GAN model outperforms the baseline methods by over 30.54% and has apparent advantages in multi-step prediction.

INDEX TERMS Intelligent transport system, GCN-GAN, graph machine learning, time series prediction.

I. INTRODUCTION

Due to the development of technology and the economy, the number of vehicles on the road grows each year, leading to urban congestion. At the same time, the incomplete urban road planning and inadequate traffic facilities have exacerbated this situation, resulting in a huge amount of energy waste and pollution emissions [1]. Many methods have been proposed to solve the problem of traffic congestion, such as increasing infrastructure construction [2], increasing the number of traffic officers [3] and restricting travel according to the license plate number [4], [5]. Compared with investing a lot of costs in control, if we can predict the traffic flow in advance, the occurrence of traffic congestion can be considerably avoided. Thus, it is necessary to propose an accurate prediction method suitable for modern governance.

Due to the disruptive impact of computing and communication in the field of transportation, several professionals jointly proposed the term Intelligent Transportation Systems (ITS) in the 1980s [6]. ITS is a system that attempts to solve various road traffic problems using information and communication technology [7]. Specifically, by integrating sensors, traffic signals, and personnel information, ITS can achieve precise prediction and control of traffic. In the process of using ITS, traffic prediction can effectively improve the effect of traffic information regulation [8].

Time series prediction is a method to extract valuable information and predict the next trend of the system by analyzing the past data [9]. As one of the typical time series problems, urban traffic flow forecast is essential in ITS. For example, driving routes can be dynamically planned

The associate editor coordinating the review of this manuscript and approving it for publication was Joey Tianyi Zhou.

by predicting traffic flow, thus effectively reducing traffic congestion. Randomness, periodicity, and Spatio-temporal characteristics are the challenges in traffic flow forecast. In addition, the deepening of urbanization also expands the urban road network and dramatically increases the computational complexity of predictions.

At present, many models have been put forward and applied in the field of traffic flow prediction. The disadvantages of these methods are insufficient prediction accuracy [10], short prediction step length [11], difficulty in parallel prediction [12], and inability to deal with irregular and unexpected traffic conditions [13].

Therefore, to obtain a model with a good prediction effect, we must overcome the above problems.

In this paper, we propose a new prediction model based on time series (GCN-GAN), which combines Graph Convolution Neural Network (GCN) and Generative Adversative Neural Network (GAN) to forecast traffic flow. The main contributions are as follows.

- GCN-GAN leverages Graph Convolution Neural Network (GCN) module and Generative Adversative Neural Network (GAN) module to predict urban traffic flow.
- GCN-GAN can generate prediction results in parallel instead of the traditional one by one, so it has obvious advantages in multi-step prediction.
- GCN-GAN model outperforms the baseline methods by over 30.54%.

Our proposed model significantly improves the accuracy of traffic flow prediction and performs well in multi-step prediction. The aggregation analysis function based on graph neural network improves the robustness of the model and reduces the negative impact of special situations in traffic flow on the prediction results. The organization of the article is as follows. In Section II, the research progress of traffic flow prediction methods is presented. In Section III, we briefly introduce basic knowledge and define the traffic flow prediction problem. In Section IV, we illustrate the specific architecture and algorithm of the GCN-GAN. In Section V, we discuss experimental design and experimental results. Future work is prospected in Section VI.

II. RELATED WORKS

In this paper, we can divide traffic flow prediction into statistical methods and machine learning methods.

A. STATISTICAL METHODS

As one of the statistical methods, the autoregressive integrated moving average model (ARIMA) is specified by various training data. We test the model at each stage with completely different data to make the model more accurate and general. The results in [14] suggest that ARIMA models trained on time-series data can achieve better results than those trained on non-time-series data. When the non-linear GARCH model is added to the ARIMA model, both the conditional mean of the traffic flow sequence and the heteroscedasticity can be calculated.

Conditional mean and conditional variance contained in the data can be predicted simultaneously by the ARIMA-GARCH model [15], thus a continuous time-varying confidence interval can be calculated. These calculated time-variant confidence intervals are more temporally deterministic than the consistent confidence intervals provided by standard ARIMA.

In addition to being suitable for traffic flow prediction, ARIMA is also effective for flow prediction in subways [16], scenic spots, and other places.

However, using the ARIMA model to achieve high-accuracy prediction of traffic flow requires a large amount of traffic data for model training. Therefore, ARIMA does not perform well when the amount of data is insufficient. Kalman filter [17] is a state-space method in the time domain, which regards the signal as the output of the linear system under the action of white noise. It has the advantages of a flexible selection factor and short prediction time. In view of the problem that the performance of the classical Kalman filter and the extended Kalman filter degrades when dealing with non-Gaussian noise, many improved Kalman filter models have been proposed in recent years [18], [19], [20]. In addition to the above models, statistical methods also include the Grey prediction method [21], [22] and Exponential smoothing method [23].

The statistical methods have the advantages of simple parameters and easy calculation. However, the model's performance highly depends on the stationarity of the data. They can not reflect the uncertainty and non-linear characteristics of dynamic traffic flow and can not overcome the influence of random disturbance factors.

B. MACHINE LEARNING METHODS

In the field of Machine Learning Methods, Castro-Neto *et al.* [24] applied Support Vector Regression (SVR) to the field of traffic prediction as early as 2009.

Luo *et al.* [25] proposed a high-accuracy predictive model that combines SVR and Discrete Fourier Transform (DFT). Compared with ARIMA, EMD-SVR and other models, DFT-SVR has outstanding performance in short-term prediction. Due to the development of computing power in recent years, deep learning [26], as a new non-linear method, has attracted great attention and use by researchers and business people. A Deep learning network is a complex perceptron with multiple layers, each containing a large number of neurons. It implements the complex calculation by learning the weights in the non-linear network structure and finally realizes the purpose of high-precision traffic flow prediction. Kumar *et al.* [27] applied ANN to achieve short-term predictions for the future based on traffic data from past periods. Their experimental results indicated that when the time interval of traffic flow prediction is increased to 300% of the original, the prediction accuracy of the neural network remains consistent.

To realize the prediction of traffic flow in different time steps, Chen *et al.* [28] added Ensemble Ensemble Empirical Mode Decomposition (EEMD) on the basis of Artificial Neural Network (ANN). They found that this ensemble framework model significantly outperformed classical neural network models in prediction. However, these methods are mainly single-step predictions, which can not effectively avoid cumulative error in the multi-step forecast. To solve this problem, the time series prediction method incorporating long short-term memory network (LSTM) has achieved considerable results [29], [30], [31]. However, LSTM has disadvantages (1) limited historical data mining ability and (2) low computational efficiency. More importantly, the above model only considers the temporal characteristics of traffic data but ignores the spatial features, which will undoubtedly lead to the model's prediction not being restricted by the standard urban traffic structure, thus reducing the accuracy of the prediction results. In addition, the prediction methods based on Graph Convolutional Neural Network are also gradually applied in traffic.

Zhao *et al.* [32] proposed the method of constructing a neural network from the graph and used it in traffic flow forecasting for the first time. This model is called Temporal Graph Convolutional Network (T-GCN). It innovatively adds a gated recurrent unit (GRU) structure to the graph convolutional network (GCN). Experiments show that the spatiotemporal correlations implicit in traffic data can be well captured and learned by the T-GCN model. The trained model outperforms all previously proposed baseline methods in the prediction results on real-world traffic datasets. T-GCN tries to combine spatial features and temporal features for prediction and has made significant progress, but there is still room for improvement in prediction accuracy. The features and limitations of the above-mentioned methods are summarized in Table 1.

III. PRELIMINARY

A. GRAPH CONVOLUTIONAL NEURAL NETWORK (GCN)

Traditional deep learning neural networks (DNN) cannot represent the relational data of vertices and edges. Hence, graph neural network (GNN) appears to solve this graph data representation problem.

The process of GNN can be divided into two steps: the first step is the propagation process, which refers to the updating of nodes over time. The second step is the output process, which is obtaining the target output (such as the category of each node) based on the final node representation. GNN learning is achieved by Almeida-Pineda algorithm [33]. The characteristic of the algorithm is that the whole graph converges through the propagation process, and then the corresponding gradient is calculated on the convergent solution. This way, we do not need to store the intermediate states needed for the gradient calculation process. But the mapping of the whole graph must be compressed to ensure a convergent solution for the propagation process. The disadvantages of GNN are as follows: (1) The fixed point hidden state update method is very inefficient. (2) Using the same parameters in the process of iteration. (3) Some edge information features cannot be effectively modelled.

GCN belongs to a kind of GNN. GCN differs from ordinary GNN because it introduces a convolution function to learn by extracting spatial features. The most significant innovation of GCN based on GNN lies in using a convolution operator for information aggregation.

1) BASIC KNOWLEDGE

As a linear function transform, Fourier transform [34] can convert the signal between the time domain and the frequency domain. The specific formula is as follows.

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x u} dx \tag{1}$$

where the independent variable x represents the time in seconds, and the transformation variable u represents frequency (to Hertz units).

We define an acyclic graph G with vertices N, its Laplacian matrix M_L can be defined as $M_L = -M_A + M_D$, where M_D is the degree (in-degree and out-degree) matrix, and M_A is the adjacency matrix. The specific calculation formula of the element of M_L is as follows:

$$M_{L_{j,i}} = \begin{cases} -1 & \text{while } j \neq i \;\&\&\; v_j \text{ is contiguous with } v_i \\ \deg\left(v_j\right) & \text{while } j = i \\ 0 & \text{else} \end{cases} \tag{2}$$

2) CORE CONCEPT OF GCN

The GCN model [35] usually generates a new node representation by aggregating the node itself and the surrounding information of the node.

After essential spectrum convolution and Layer-wise linear model processing [36], the expression of GNN is as follows:

$$M_H^{(l+1)} = \sigma\left(\hat{M}_D{}^{-\frac{1}{2}} \hat{M}_A \hat{M}_D{}^{-\frac{1}{2}} M_H^{(l)} M_W^{(l)}\right) \tag{3}$$

In the above formula, the input of layer l network is $M_H^l \in \mathbb{R}^{\mathbb{N}\times\mathbb{D}}$ (initial input is $M_H(0) = X$), and D represents the dimension of each node vector. X represent the matrices input into the GCN. $\hat{M}_A = M_A + I_N$ is the self-joined adjacent matrix, $\hat{M}_D$ is the degree matrix of $\hat{M}_A$, $M_W^{(l)} \in \mathbb{R}^{D\times D}$is the training parameter.

B. GENERATIVE ADVERSARIAL NETWORK (GAN)

In 2014, Generative Adversarial Network (GAN) [37] was proposed by Goodfellow and has been widely applied in the field of computer vision. The basic idea of GAN is that the inputs (randomly distributed vectors) pass through a generator composed of neural networks to generate structured high-dimensional data. When the GAN network is trained, the discriminant network will continuously improve the recognition ability. In contrast, the generative network will continuously improve the generative ability and reduce the discriminant network's discriminant ability. In the process

TABLE 1. Two major approaches for traffic flow prediction.

	Methods	Features	Limitations
STATISTICAL METHODS	•ARIMA •Kalman filter •Grey prediction method •Exponential smoothing method	• The model parameters are simple, and no additional variables need to be added. • They can solve the problem of traffic flow change in different time and period to a certain extent.	• The time series data is required to be stable. • To achieve high-accuracy prediction of traffic flow requires a large amount of traffic data for model training. • They can not reflect the uncertainty and non-linear characteristics of dynamic traffic flow and can not overcome the influence of random disturbance factors.
MACHINE LEARNING METHODS	•SVR •ANN •LSTM •T-GCN	• Machine learning methods can learn dynamic characteristics during traffic and predict possible emergencies. • They are more powerful than the traditional statistical methods and can efficiently capture the spatial and temporal information of the traffic system.	• The high precision parallel prediction is not well realized. • A large amount of data is needed for training, and the generalization ability of the model depends on the quality of the data. •The ability to aggregate spatial information still needs to be strengthened.

of competition between the two networks, the ability of GAN to generate new samples will be improved.

1) DEFINITION OF GENERATIVE AND ADVERSARIAL PROBLEM

The probability density function of the target high-dimensional data is $T_{data}(x)$, and the probability density function of the Generator is set as $T_G(x;\theta)$, where θ is the parameter of the learned distribution. By optimizing θ, let $T_G(x;\theta)$ is infinitely close to $T_{data}(x)$. Our purpose is to extract m data $x^1, x^2, x^3 \ldots x^m$ from $T_{data}(x)$, and then optimize parameter $L = \prod_{i=1}^{m} T_G\left(x^i;\theta\right)$ to maximize the maximum likelihood function L. In fact, it is classified as the problem of calculating the minimum KL distance between $T_{data}(x)$ and $T_G(x;\theta)$.

2) GENERATOR AND DISCRIMINATOR

Generator is a neural network whose goal is to find the set of parameters θ that minimizes the distance between $T_{data}(x)$ and $T_G(x;\theta)$.

$$G^* = \arg\min_G \mathrm{Div}\left(T_G(x;\theta), T_{data}(x)\right) \tag{4}$$

The goal of the discriminator is to be a "quality inspector" by distinguishing as much as possible between real data from the dataset and mock data from the generator, but also to improve performance in correcting errors.

$$D^* = \arg\max_D V(D,G) \tag{5}$$

Assuming G is a fixed value in the above formula, then:

$$\begin{aligned} V &= E_{x\sim T_{\text{data}}} \log D(x) + E_{x\sim T_G} \log(-D(x)+1) \\ &= \int_x \left[T_{\text{data}}(x)\log D(x) + T_G(x,\theta)\log(-D(x)+1)\right] dx \end{aligned} \tag{6}$$

If we want to find the best discriminator D, we need to maximize the following formula

$$T_{\text{data}}(x)\log D(x) + T_G(x,\theta)\log(-D(x)+1) \tag{7}$$

C. RNN, LSTM AND BI-LSTM

The time series prediction task needs to handle serialization information well, which means our model needs to capture the relationship between the inputs in the time series. The recurrent neural network (RNN) [38] appears to solve the time series prediction problem. However, since backpropagation is based on the chain rule principle, RNNs exhibit the hidden dangers of vanishing gradients and exploding gradients with the increasing complexity of neural networks [39]. Long Short-Term Memory (LSTM) [40] is a special RNN (threshold RNN), which adds threshold selection to solve the gradient problem encountered in long sequence training. LSTM has better performance than common RNN models on longer sequence predictions. LSTM consists of a threshold for input, an output threshold, and a forgetting threshold, establishing a self-loop connection. The LSTM memory unit can remember values at any time interval and control the flow of information inside and outside the unit through three thresholds. LSTM networks are suitable for extracting features and making predictions from time series data of unknown length. The classical LSTM model formula is as follows:

$$\begin{aligned} f_t &= \sigma\left(z_f\right) \\ i_t &= \sigma\left(z_i\right) \\ o_t &= \sigma\left(z_0\right) \\ c_t &= \tanh(z) \odot i_t + c_{t-1} \odot f_t \\ h_t &= \tanh\left(c_t\right) \odot o_t \end{aligned} \tag{8}$$

The unavoidable problem of the traditional RNN model and LSTM model (Figure 1) is that information can only be

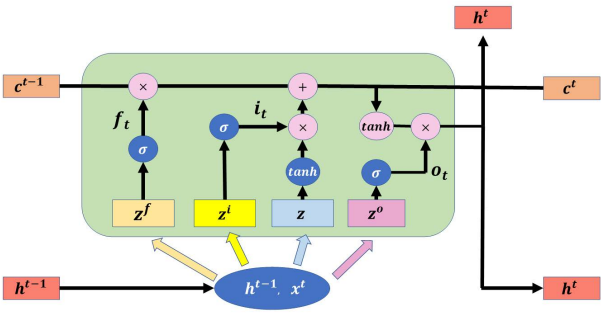

FIGURE 1. Internal structure of LSTM.

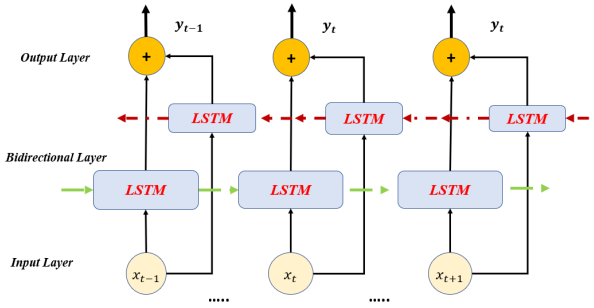

FIGURE 2. Bi-directional LSTM model.

propagated forward. That is to say, the state of time t only depends on the previous sequence information.

Bidirectional Long Short-Term Memory (Bi-LSTM) [41] is widely used in sequence processing to capture the information before and after a sequence. It is obtained by combining forward-propagated LSTM and backward-propagated LSTM, which can analyze sequence information equally (Figure 2).

D. TRAFFIC FLOW PREDICTION PROBLEM

Traffic flow prediction is a typical problem of Spatio-temporal data prediction. Different kinds of traffic data are embedded in continuous and dynamically changing space and time. Therefore, obtaining reasonable time and space features from past data is the key to achieving accurate prediction. The specific problem of traffic flow prediction is defined as follows: we know the traffic graph $G = (V, E)$ (where the node set is represented by V, and the edge set is represented by E) and the historical SE (Source-End) matrices of urban traffic flow ($Flow_{his} = Flow_1, Flow_2, \ldots, Flow_t$), which are the source point-end vehicle flow matrices, and carry out multi-intersection urban traffic flow prediction through $Flow_{his}$ information. For each node, an eigenvector of fixed length is generated every minute as follows.

$$Flow_{t+k} = \text{Predict } [Flow_{his}] \tag{9}$$

where $k(k >= 1)$ is the step size of prediction, *Predict* is the prediction method we need to seek.

IV. METHOD

The traditional traffic flow forecast is a single-step forecast, and the result is not ideal. A hybrid model network structure (GCN-GAN) based on GCN and GAN is proposed in this section to improve the accuracy of traffic flow prediction and increase the step length of traffic flow prediction. GCN-GAN is designed to jointly predict the traffic flow at multiple intersections within a single step or multiple steps (as shown in Figure 3). The general idea is to apply graph convolution directly to historical traffic flow data $Flow_{his}$ (graph structure data represented by matrix). GCN extracts patterns and features in the frequency domain, and the extracted information is used for time series prediction.

It is complicated to calculate with graph information, so we convert it into SE matrices of traffic flow. In the SE matrix, when the nodes are connected, the position corresponding to a particular row and column is marked as 1. Otherwise, it is marked as 0. Then a filter is constructed in the Fourier domain, and GCN is used to extract the spatial information features of multiple urban traffic intersections and their adjacent regions in the SE matrix.

$$f\left(M_H^{(l)}, M_A\right) = \sigma\left(\hat{M}_D^{-\frac{1}{2}}\hat{M}_A\hat{M}_D^{-\frac{1}{2}}M_H^{(l)}M_W^{(l)}\right) \tag{10}$$

In the above formula, $M_W^{(l)}$ is the parameter matrix of the l layer GCN neural network for traffic information extraction, which will be optimized in each iteration.

A. GAN STRUCTURE

The design of the GAN network in this section is inspired by Conditional Generative Adversarial Nets (C-GAN) [42]. The structure contains Generator and Discriminator, respectively.

1) GENERATOR DESIGN

The generator is internally composed of DNN, as shown in Figure 5, and the input conditions are:

- History traffic flow SE matrices $Flow_1, Flow_2, \ldots, Flow_t$
- $Flow_{gcn_1}, Flow_{gcn_2}, \ldots$ and $Flow_{gcn_t}$ are obtained by feature extraction of historical traffic flow SE matrices using GCN
- The step size k that needs to be predicted
- Normally distributed data z

Here, we input the SE matrices of historical traffic flow $Flow_1, Flow_2, \ldots, Flow_t$ and the matrices extracted by GCN $Flow_{gcn_1}, Flow_{gcn_2}, \ldots, Flow_{gcn_t}$ into the generator together. Our idea is based on residual connection [43], aiming to combine linear and nonlinear features in traffic flow data. The generator needs to find a parameter set θ that minimizes the distance between $T_G(x;\theta)$ and $T_{data}(x)$ under the conditions of historical data, graph convolution and prediction steps.

$$G^* = \arg\min_G \text{Div}\left(T_G(x,\theta), T_{data}(x)\right) \tag{11}$$

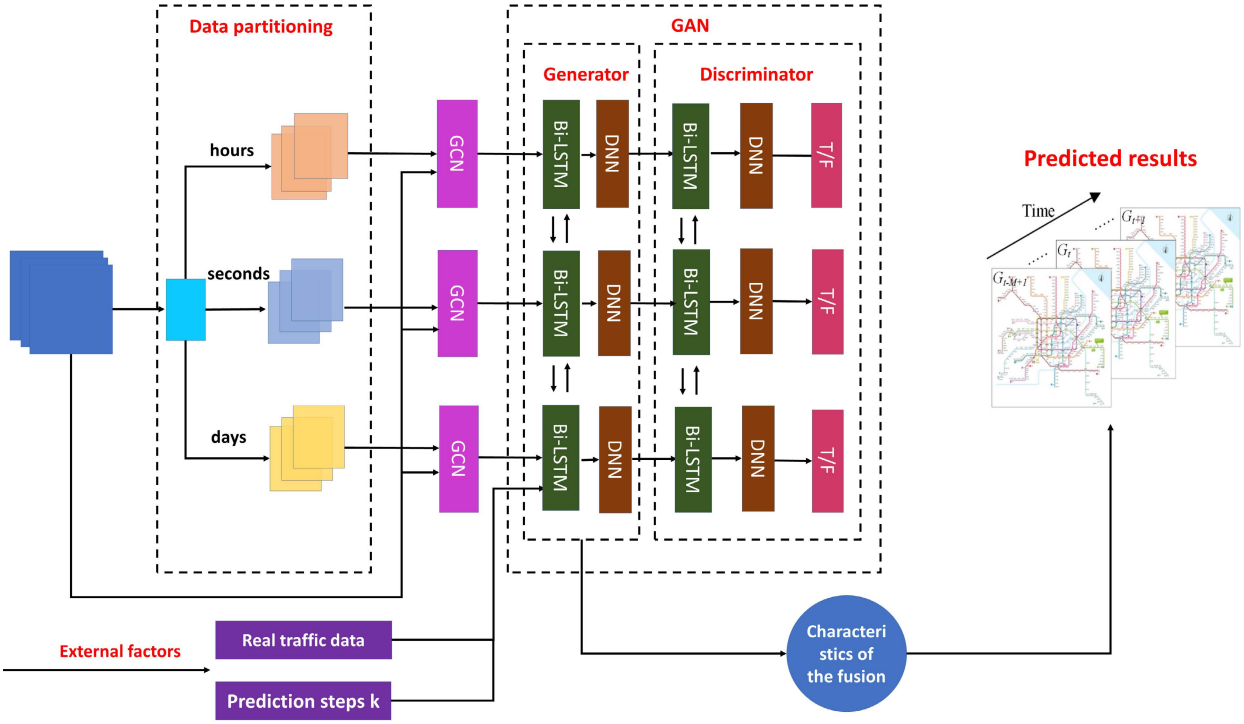

FIGURE 3. GCN-GAN structure.

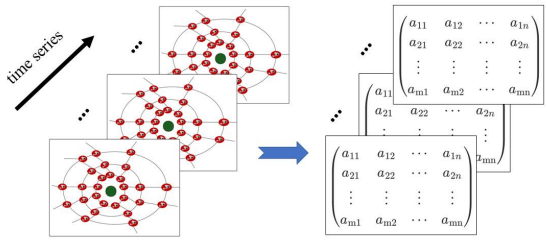

FIGURE 4. Convolution diagram of urban traffic flow according to time series.

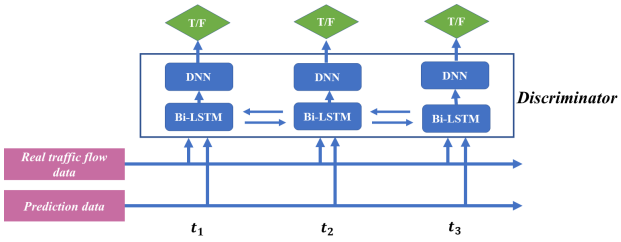

FIGURE 6. Discriminator network structure.

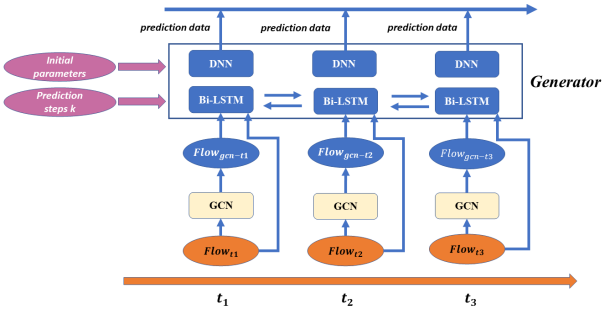

FIGURE 5. Generator network structure.

B. DISCRIMINATOR DESIGN

In the GAN model designed in this paper, the real and predicted data are input into the discriminator, and the discriminator outputs the judgment result of the input data. The discriminator distinguishes between the generated data $T_G(x;\theta)$ and the real data $T_{data}(x)$. The score of real data tends to be 1, and the score of generated data tends to be 0.

$$T_{data}(x)\log D(x) + T_G(x,\theta)\log(-D(x)+1) \qquad (12)$$

The Algorithm process of GCN-GAN is shown as Algorithm 1.

V. EXPERIMENTS

A. DataSet

We used surveillance video data at 27 intersections in a city in China within two months. Due to the influence of irresistible factors, the video resolution obtained at each intersection has a certain degree of acceptable difference, and the full size of our data is about 1.3TB. Image frames were extracted from the video at the frequency of two frames per second, and the resolution of image frames was modified to 128*128, which were named according to the intersection number and time. The traffic flow time-series image data set contains 34.99 million images. After target detection and

Algorithm 1 The Algorithm Process of GCN-GAN

Input: Urban traffic SE matrices of multi-intersection traffic flow $Flow_1, \ldots, Flow_t$

Output: Traffic flow prediction result matrix $Flow_{t+k}$

1: Use double-layer GCN to capture position information in SE matrices to get $Flow_{gcn_1}, \ldots, Flow_{gcn_t}$ based on Equation 10

2: Initialize state generator G and discriminator D

3: Feed $Flow_{gcn_1}, \ldots, Flow_{gcn_t}$, $Flow_1, \ldots, Flow_t$, number of prediction steps k ($k = 1, 3, 5$), normal distribution $z\,(3, 0.5^2)$ into generator G

4: **repeat**

5: Extract m data $x^1, x^2, x^3 \ldots x^m$ from $T_{data}(x)$

6: Fix G, optimize D by computing the gradient of D

$$\nabla_{\theta_d} \frac{1}{m} \int_x \left[T_{\text{data}}(x) \log D(x) + T_G(x, \theta) \log(1 - D(x)) \right]$$

7: Fix D, optimize G by computing the gradient of G

$$\nabla_{\theta g} \frac{1}{m} \int_x \left[T_G(x, \theta) \log(-D(x) + 1) \right]$$

8: **until** The gradient of the D is approximately equal to 0

9: End of training

10: Convert the new $Flow_1, \ldots, Flow_t$ to $Flow_{gcn_1}, \ldots, Flow_{gcn_t}$, and input them into the generator G together with the number of predicted steps k, and output the predicted traffic flow.

text information extraction, a real-world multi-road traffic Flow time-series text data set (Multi-Road Traffic Dataset, MTD) is generated. The vehicle types in the MTD dataset include trucks, cars, and buses. The MTD dataset is designed in Jason-type text format, and the MTD of each intersection contains the following field information (as shown in Table 2).

For the obtained MTD dataset, we first calculate the traffic flow of all 27 intersections in units of minutes to generate the traffic flow SE matrices *Flow*. For the minute-by-minute data at each intersection for a month, there are 43200 SE matrices, that is, $T_m = 43200$. In the experimental process, 5-fold cross-validation is used to obtain model data and avoid the influence of algorithm randomness to the greatest extent.

B. PARAMETER SETTINGS

Our experimental environment is listed here: the software environment is PyCharm 2022.1.2; The deep learning framework used is TensorFlow2.0.0. The operating system is Win10. The training device was dictated by the 11th Gen Intel(R) Core(TM) i7-11800h @ 2.30GHz. The graphics card configuration used for the calculations is the Tesla V100. The prediction accuracy will inevitably decrease gradually with the increase of the prediction step. To ensure the better practicability of our model, we set the prediction step k as 1, 3, and 5, which were compared with the baseline algorithm.

TABLE 2. MTD dataset field information.

Field Name	Description
cross_roadid	intersection number
West2East_W_Straight	Go straight from west2east (west side)
West2East_E_Straight	Go straight from west2east (east side)
West2East_W_Left	Turn left from west2east (turn left on the west side)
West2East_W_Right	Turn right from west2east (turn right on the west side)
East2West_W_Straight	Go straight from east2west (west side)
East2West_E_Straight	Go straight from east2west (east side)
East2West_E_Left	Turn left from east2west (turn left on the east side)
East2West_E_Right	Turn right from east2west (turn right on the east side)
North2South_N_Straight	Go straight from north2south (north side)
North2South_S_Straight	Go straight from north2south (south side)
North2South_N_Left	Turn left from north2south (turn left on the north side)
North2South_N_Right	Turn right from north2south (turn right on the north side)
South2North_N_Straight	Go straight from South2North (north side)
South2North_S_Straight	Go straight from South2Northt (south side)
South2North_S_Left	Turn left from south2north (turn left on the south side)
South2North_S_Right	Turn right from south2north (turn right on the south side)

The dimension of the Bi-LSTM of the generator is designed to be 27*27. The number of layers of the DNN is 3, so the dimension of the DNN is $(27 \times 128) \times (128 \times 128) \times (128 \times k)$. Depending on the value of k, the generator generates prediction data of dimension $(1 \times k)$ at a point in time. In the discriminator, we set the dimension of Bi-LSTM to be 27×27. The number of layers of the DNN is three layers, so the dimension is $(27 \times 128) \times (128 \times 64) \times (64 \times 1)$, and the activation function of the last layer is Sigmoid. The discriminator outputs a score for judging the authenticity of the data.

C. BASELINE ALGORITHMS

This section uses ARIMA, SVR, DNN, and LSTM as the baseline algorithms for comparative experiments.

- Autoregressive Integrated Moving Average model (ARIMA) [44]:
  The basic idea of ARIMA is to use the features of current and past moments in a time series to predict possible future values.
- Support Vactor Regression (SVR) [45]: The ultimate goal of SVR optimization is to minimize the maximum distance between the sample points and the hyperplane so that the model's predicted value is as close to the true value as possible and has excellent generalization ability in the face of unknown data.
- Deep Neural Network (DNN) [46]: A Deep neural network realizes complex prediction and calculation through connections and nesting among neurons.
- Long short-term memory (LSTM) [47]: LSTM consists of a forgetting mechanism, an input mechanism, and an output mechanism and is mainly used to solve the problems of memory retention and gradient disappearance in

TABLE 3. Prediction results of traffic flow at multiple intersections when the step size $k = 1$.

Algorithm	MAE_{avg}	MSE_{avg}	$RMSE_{avg}$
ARIMA [39]	18.16	774.51	27.83
SVR [40]	14.32	504.90	22.47
DNN [41]	14.01	398.80	19.97
LSTM [42]	13.55	359.10	18.95
T-GCN [32]	11.20	266.02	16.31
GCN-GAN	**10.44**	**240.87**	**15.52**

TABLE 4. The improvement of SVR, DNN, LSTM and GCN-GAN in prediction results compared with ARIMA when the step size $k = 1$.

Algorithm	MAE_{avg}	MSE_{avg}	$RMSE_{avg}$
ARIMA [39]	0%	0%	0%
SVR [40]	21.15%	34.81%	19.26%
DNN [41]	22.85%	48.51%	28.24%
LSTM [42]	25.39%	53.64%	31.91%
T-GCN [32]	38.32%	65.65%	41.39%
GCN-GAN	**44.93%**	**68.90%**	**44.23%**

TABLE 5. Prediction results of traffic flow at multiple intersections when the step size $k = 3$.

Algorithm	MAE_{avg}	MSE_{avg}	$RMSE_{avg}$
ARIMA [39]	24.60	1246.09	35.30
SVR [40]	19.95	830.02	28.81
DNN [41]	19.58	665.12	25.79
LSTM [42]	19.02	602.70	24.55
T-GCN [32]	17.74	454.12	21.31
GCN-GAN	**15.04**	**339.66**	**18.43**

TABLE 6. The improvement of SVR, DNN, LSTM and GCN-GAN in prediction results compared with ARIMA when the step size $k = 3$.

Algorithm	MAE_{avg}	MSE_{avg}	$RMSE_{avg}$
ARIMA [39]	0%	0%	0%
SVR [40]	18.90%	33.38%	18.39%
DNN [41]	20.41%	46.62%	26.94%
LSTM [42]	22.68%	51.63%	30.45%
T-GCN [32]	27.89%	63.56%	41.39%
GCN-GAN	**38.86%**	**72.74%**	**47.79%**

TABLE 7. Prediction results of traffic flow at multiple intersections when the step size $k = 5$.

Algorithm	MAE_{avg}	MSE_{avg}	$RMSE_{avg}$
ARIMA [39]	40.70	2713.37	52.09
SVR [40]	34.14	1781.68	42.21
DNN [41]	33.62	1489.96	38.60
LSTM [42]	32.83	1348.36	36.72
T-GCN [32]	24.14	986.59	31.41
GCN-GAN	**20.27**	**724.69**	**26.92**

long-term sequence training. Through practical parameter design, traffic flow can be effectively predicted.

- Temporal Graph Convolutional Network (T-GCN) [32]: T-GCN innovatively adds a gated recurrent unit (GRU) structure to the graph convolutional network (GCN). The model can learn the temporal and spatial characteristics of traffic data.

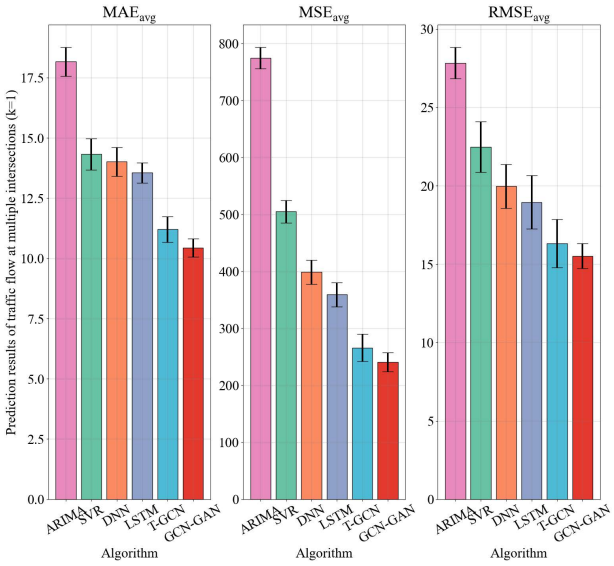

FIGURE 7. Prediction results of traffic flow at multiple intersections when the step size $k = 1$.

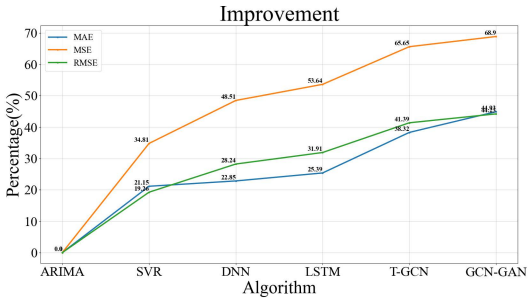

FIGURE 8. The improvement of SVR, DNN, LSTM and GCN-GAN in prediction results compared with ARIMA when the step size $k = 1$.

TABLE 8. The improvement of SVR, DNN, LSTM and GCN-GAN in prediction results compared with ARIMA when the step size $k = 5$.

Algorithm	MAE_{avg}	MSE_{avg}	$RMSE_{avg}$
ARIMA [39]	0%	0%	0%
SVR [40]	16.12%	34.34%	18.97%
DNN [41]	17.40%	45.09%	25.90%
LSTM [42]	19.34%	50.31%	29.51%
T-GCN [32]	40.69%	63.64%	39.70%
GCN-GAN	**50.20%**	**73.29%**	**48.32%**

D. EXPERIMENT PROCESS

We input the training data $Flow_1, \ldots, Flow_t$ into the GCN-GAN network and obtain the $Flow_{gcn_1}, \ldots, Flow_{gcn_t}$ through GCN. Feed the training data, $Flow_{gcn_1}, \ldots, Flow_{gcn_t}$, and k value into the generator. We conducted three sets of experiments under three prediction steps ($k = 1, 3, 5$), and

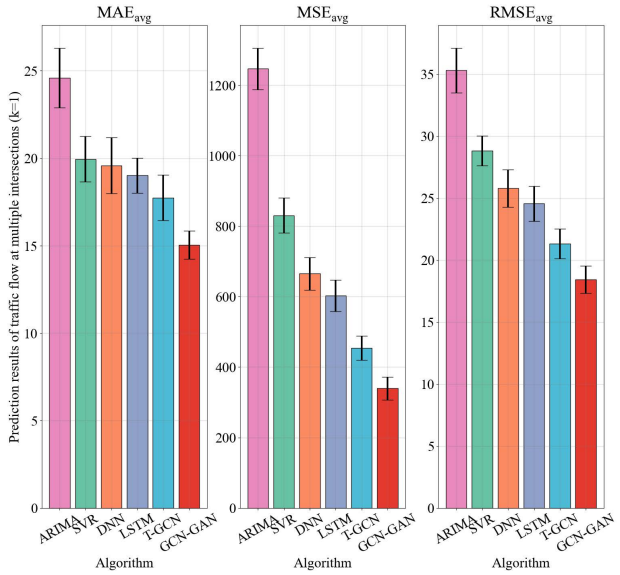

FIGURE 9. Prediction results of traffic flow at multiple intersections when the step size $k = 3$.

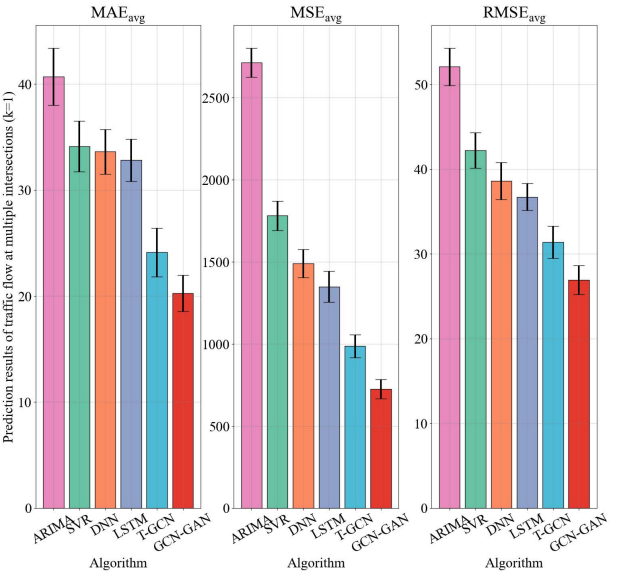

FIGURE 11. Prediction results of traffic flow at multiple intersections when the step size $k = 5$.

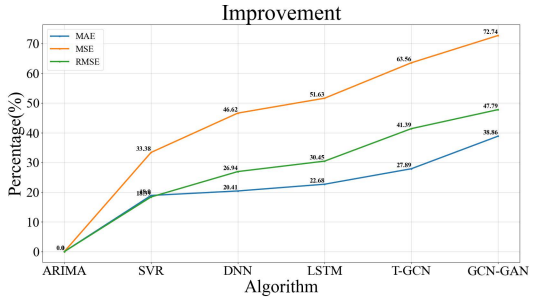

FIGURE 10. The improvement of SVR, DNN, LSTM and GCN-GAN in prediction results compared with ARIMA when the step size $k = 3$.

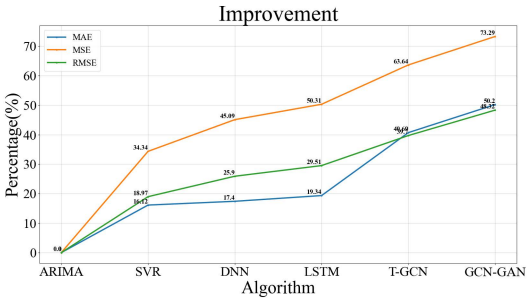

FIGURE 12. The improvement of SVR, DNN, LSTM and GCN-GAN in prediction results compared with ARIMA when the step size $k = 5$.

TABLE 9. The improvement of GCN-GAN's prediction result compared with the average prediction results of the other four baseline algorithms under three step sizes.

Index	Step $k = 1$	Step $k = 3$	Step $k = 5$	Average
MAE_{avg}	26.73%	25.46%	38.74%	**30.31%**
MSE_{avg}	47.71%	55.28%	56.45%	**53.15%**
$RMSE_{avg}$	26.47%	32.12%	33.04%	**30.54%**

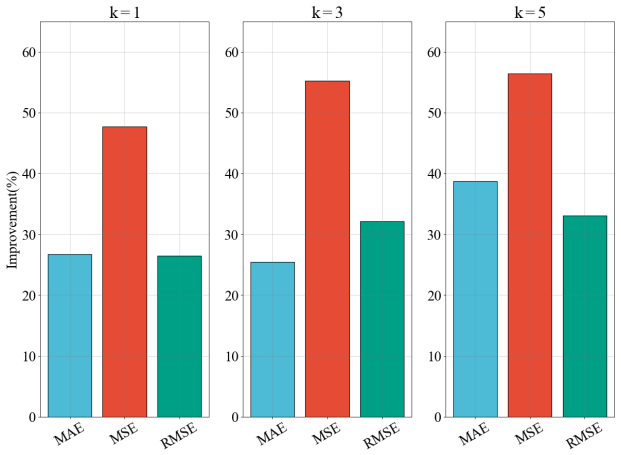

FIGURE 13. The improvement of GCN-GAN's prediction result compared with the average prediction results of the other four baseline algorithms under three step sizes.

each group was trained for epoch=10000 times. Training and testing are based on the principle of 5-fold cross-validation.

E. EVALUATION METRICS

Some evaluation metrics should be selected to measure the discrepancy between the predicted results and observed values to evaluate the GCN-GAN model performance for traffic flow. In general, we specify the model predictive results as $x^{(1)}, x^{(2)}, x^{(3)} \ldots, x^{(m)}$, the true value is $y^{(1)}, y^{(2)}, y^{(3)} \ldots, y^{(m)}$, the prediction function is $h(x)$. The m value here is determined by the 5-fold cross-validation. In this paper, we use the following metrics.

1) MEAN ABSOLUTE ERROR (MAE)

As one of the most classic regression loss functions, MAE represents the mean of the absolute error between the true

TABLE 10. Symbol notation and definition.

Symbol	Description
x	The time domain variable in the Fourier formula
u	The frequency domain variable in the Fourier formula
M_L	Laplace matrix of traffic flow graph
M_A	Adjacency matrix of traffic flow graph
M_D	Degree (in-degree and out-degree) matrix of traffic flow graph
N	Number of nodes in the graph
D	The dimension of each node vector
$\hat{M}_A$	Self-joined adjacent matrix of traffic flow graph
$\hat{M}_D$	The degree matrix of $\hat{M}_A$
$M_H^{(l)}$	The input of layer l network (initial input $M_H^{(0)} = Flow_i$)
$M_W^{(l)}$	The parameter matrix of the l layer GCN neural network for traffic information extraction
$Flow_i$	History traffic flow SE matrices ($i = 1, ...t$)
$Flow_{gcn\_i}$	Features extracted from SE matrices of historical traffic flow using GCN ($i = 1, ..., t$)
k	Step size ($k = 1, 3, 5$)
z	Normally distributed data
$T_{data}(x)$	The probability density function of the target high-dimensional data
$T_G(x, \theta)$	The probability density function of the Generator

value and the observed value.

$$MAE(X, h) = \frac{1}{m}\sum_{i=1}^{m}|h(x^{(i)}) - y^{(i)}| \tag{13}$$

2) MEAN SQUARED ERROR (MSE)

MSE calculates the average value of squared errors between predicted and true values. MSE can measure the gap between different data to some extent. The higher the accuracy of the model to experimental data, the smaller the value of MSE.

$$MSE(X, h) = \frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2 \tag{14}$$

3) ROOT MEAN SQUARED ERROR (RMSE)

RMSE is the square root of the mean squared deviation of the predicted value from the true value over n predictions. Through RMSE, we can visually observe the deviation between the actual and predicted value.

$$RMSE(X, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2} \tag{15}$$

where $x^{(i)}$ represents one of the 27 multi-roads' traffic flow.

According to the error formula, we can calculate the respective averages of the three metrics at each intersection under 5-fold cross-validation. Meanwhile, we need complete SE matrices for the model outcome. Thus, we compute the accumulation average for 27 multi-roads MAE, MSE, and RMSE.

$$MAE_{avg} = \frac{\sum_{i=1}^{27} MAE_i}{27} \tag{16}$$

$$MSE_{avg} = \frac{\sum_{i=1}^{27} MSE_i}{27} \tag{17}$$

$$RMSE_{avg} = \frac{\sum_{i=1}^{27} RMSE_i}{27} \tag{18}$$

F. RESULTS AND DISCUSSION

To better demonstrate the superiority of the GCN-GAN, we make predictions based on MTD test sets under three step sizes ($k = 1, 3, 5$). By computing the MAE_{avg}, MSE_{avg} and $RMSE_{avg}$ of the 27 intersections (as shown in Table 3, Table 5 and Table 7), we can see that GCN-GAN has a significant improvement in the overall prediction effect. To demonstrate the superiority of our model in traffic flow prediction more clearly, we take the ARIMA algorithm as the benchmark to calculate the improvement of SVR, DNN, LSTM, and GCN-GAN in prediction results compared with ARIMA (as shown in Table 4, Table 6, Table 8). The above prediction results and improvement are visualized for further analysis (as shown in Figure 7, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12).

As shown in Table 3, Table 5 and Table 7, regardless of the prediction step size, GCN-GAN achieves better performance than other baseline algorithms on all four indexes. Since MSE is more sensitive to the fluctuation of outliers than MSE, the value of MSE_{avg} is much larger than MAE_{avg} and MSE_{avg} in terms of error. MAE_{avg} showed the intuitive accuracy of GAN-GAN algorithm's prediction, which reached the lowest values of 10.44 ($k = 1$), 15.04 ($k = 3$) and 20.27 ($k = 5$) in the three-step sizes. With the increase in predicted step size, the results of MSE_{avg} and $RMSE_{avg}$ revealed the stability of GCN-GAN. While the improvement of other baseline algorithms decreased with the increase of predicted step size, the improvement of GCN-GAN still increased. When k=5, the improvement of RMSE was 48.32%. Although the prediction accuracy will decrease with the step size increase for all prediction models, GCN-GAN shows the best characteristics compared with other baseline methods. The decline rate is slow and stable for GCN-GAN.

To more intuitively show the influence of step size changes on the prediction error of the GCN-GAN model, we calculated the average error of ARIMA, SVR, DNN, LSTM and T-GCN algorithms under three steps and further calculated the improvement of the GCN-GAN's prediction error relative to the average error (as shown in Table 9). As shown in Figure 13, with the increase of the predicted step length, the overall improvement of the three indicators of the GCN-GAN model is gradually improving, even though the absolute error is increasing. This phenomenon shows that our model performs better than other baseline algorithms in multi-step prediction. That is, the rate of accuracy decline is relatively gentle and stable. It is worth noting that GCN-GAN's prediction of the traffic flow at multiple intersections generates the

traffic flow at all intersections at one time instead of running a prediction model program at each intersection in turn or at the same time greatly accelerates the prediction speed.

VI. CONCLUSION

Aiming at the problem that most of the traditional traffic flow prediction methods are single-step prediction models and the prediction accuracy is low, we propose a novel traffic flow prediction model GCN-GAN, combining the Graph Convolution Neural Network (GCN) module and Generative Adversative Neural Network (GAN) module to predict urban traffic flow in this paper. It is a multi-step prediction of traffic flow at multi-intersections.

Compared with ARIMA, SVR, DNN and LSTM, we find that the GCN-GAN model has an obvious advantage in multi-step prediction, and its prediction performance is about 30.54% (average value of $RMSE_{avg}$) higher than the benchmark time series prediction model.

Our method first uses GCN to extract spatial features of traffic data and further utilizes GAN and Bi-LSTM structures for time series prediction. Due to the consideration of the spatial topological relationship between intersections in the whole traffic network, GCN-GAN can achieve more accurate results than the traditional timing prediction methods. On the other hand, it can be seen from Table 9 and Figure 13 that with the increase of the prediction step, the improvement of the average value of the GCN-GAN model compared with other baseline methods does not decrease but increases. This situation shows that the model still has specific stability when the prediction steps size increases. We mainly solve the problem of low accuracy of multi-step prediction in traffic flow prediction and provide a new idea for Spatio-temporal data integration.

This promising result will encourage us to continue to use the model for large-scale traffic flow prediction problems.

In the following study, we will consider adding a self-attention mechanism to the GCN-GAN model to achieve more optimized prediction results for timing information. Meanwhile, the experimental results are based on MTD data sets, and the superiority of the model will be verified on multiple data sets in the future.

REFERENCES

[1] A. Rahman and A. F. Hoque, "Traffic congestion in Dhaka City: Potential solutions," *Eur. J. Soc. Sci. Stud.*, vol. 2, no. 12, pp. 121–136, May 2018.

[2] D. Donaldson, "Railroads of the Raj: Estimating the impact of transportation infrastructure," *Amer. Econ. Rev.*, vol. 108, nos. 4–5, pp. 899–934, Apr. 2018.

[3] A. Dutta, S. Santra, S. Saha, A. Chakraborty, A. Kumar, A. Roy, S. Roy, D. Chakraborty, H. N. Saha, A. S. Sharma, and M. Mullick, "Intelligent traffic control system: Towards smart city," in *Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Oct. 2019, pp. 1124–1129.

[4] Z. Liu, R. Li, X. Wang, and P. Shang, "Effects of vehicle restriction policies: Analysis using license plate recognition data in Langfang, China," *Transp. Res. A, Policy Pract.*, vol. 118, pp. 89–103, Dec. 2018.

[5] W. Yao, Y. Ding, F. Xu, and S. Jin, "Analysis of cars' commuting behavior under license plate restriction policy: A case study in Hangzhou, China," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 236–241.

[6] S. Kaffash, A. T. Nguyen, and J. Zhu, "Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis," *Int. J. Prod. Econ.*, vol. 231, Jan. 2021, Art. no. 107868.

[7] H. Makino, K. Tamada, K. Sakai, and S. Kamijo, "Solutions for urban traffic issues by ITS technologies," *IATSS Res.*, vol. 42, no. 2, pp. 49–60, Jul. 2018.

[8] J. Barros, M. Araujo, and R. J. F. Rossetti, "Short-term real-time traffic prediction methods: A survey," in *Proc. Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, 2015, pp. 132–139.

[9] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 379, no. 2194, Apr. 2021, Art. no. 20200209.

[10] 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.

[11] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate multisteps traffic flow prediction based on SVM," *Math. Problems Eng.*, vol. 2013, pp. 1–8, Oct. 2013.

[12] D. Xia, M. Zhang, X. Yan, Y. Bai, Y. Zheng, Y. Li, and H. Li, "A distributed WND-LSTM model on MapReduce for short-term traffic flow prediction," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2393–2410, Apr. 2021.

[13] F. Guo, J. W. Polak, and R. Krishnan, "Predictor fusion for short-term traffic forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 90–100, Jul. 2018.

[14] H. Dong, L. Jia, X. Sun, C. Li, and Y. Qin, "Road traffic flow prediction with a time-oriented ARIMA model," in *Proc. 5th Int. Joint Conf. (INC, IMS IDC)*, 2009, pp. 1649–1652.

[15] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with ARIMA-GARCH model," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2011, pp. 607–612.

[16] D. Yan, J. Zhou, Y. Zhao, and B. Wu, "Short-term subway passenger flow prediction based on ARIMA," in *Proc. 5th IOP Conf. Ser. Earth Environ. Sci.*, Dec. 2017, pp. 464–479.

[17] G. F. Welch, "Kalman filter," in *Computer Vision: A Reference Guide*. Cham, Switzerland: Springer, 2020, pp. 1–3.

[18] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune Kalman filter for short-term traffic flow forecasting," *Phys. A, Stat. Mech. Appl.*, vol. 536, Dec. 2019, Art. no. 122601.

[19] S. Zhang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "Noise-identified Kalman filter for short-term traffic flow forecasting," in *Proc. 15th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2019, pp. 462–466.

[20] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, and J. Qin, "Hybrid dual Kalman filtering model for short-term traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 1023–1032, 2019.

[21] H. Duan and X. Xiao, "A multimode dynamic short-term traffic flow grey prediction model of high-dimension tensors," *Complexity*, vol. 2019, pp. 1–18, Jun. 2019.

[22] H. Duan, X. Xiao, and Q. Xiao, "An inertia grey discrete model and its application in short-term traffic flow prediction and state determination," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 8617–8633, Jun. 2020.

[23] H.-F. Yang, T. S. Dillon, E. Chang, and Y.-P. P. Chen, "Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 23–34, Jan. 2019.

[24] 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.

[25] X. Luo, D. Li, and S. Zhang, "Traffic flow prediction during the holidays based on DFT and SVR," *J. Sensors*, vol. 2019, pp. 1–10, Jan. 2019.

[26] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, 2021.

[27] K. Kumar, M. Parida, and V. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Proc. Social Behav. Sci.*, vol. 104, pp. 755–764, Dec. 2013.

[28] X. Chen, J. Lu, J. Zhao, Z. Qu, Y. Yang, and J. Xian, "Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network," *Sustainability*, vol. 12, no. 9, p. 3678, May 2020.

[29] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6910–6920, Nov. 2021.

[30] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *J. Adv. Transp.*, vol. 2019, pp. 1–10, Feb. 2019.

[31] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic flow prediction using LSTM with feature enhancement," *Neurocomputing*, vol. 332, pp. 320–327, Mar. 2019.
[32] L. Zhao, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2019.
[33] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
[34] R. N. Bracewell and R. N. Bracewell, *The Fourier Transform and its Applications*, vol. 31999. New York, NY, USA: McGraw-Hill, 1986.
[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–14.
[38] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
[39] A. H. Ribeiro, K. Tiels, L. A. Aguirre, and T. Schön, "Beyond exploding and vanishing gradients: Analysing RNN training using attractors and smoothness," in *Proc. 23rd AISTATS Int. Conf. Artif. Intell. Stat.*, 2020, pp. 2370–2380.
[40] G. Van Houdt, C. Mosquera, and G. Napoles, "A review on the long short-term memory model," in *Artificial Intelligence Review*, vol. 53. Berlin, Germany: Springer-Verlag, May 2020, pp. 5929–5955.
[41] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 3285–3292.
[42] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[44] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting traffic congestion using ARIMA modeling," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1227–1232.
[45] C. Li and P. Xu, "Application on traffic flow prediction of machine learning in intelligent transportation," *Neural Comput. Appl.*, vol. 33, no. 2, pp. 613–624, Jan. 2021.
[46] W. Wang, H. Zhang, T. Li, J. Guo, W. Huang, Y. Wei, and J. Cao, "An interpretable model for short term traffic flow prediction," *Math. Comput. Simul.*, vol. 171, pp. 264–278, May 2020.
[47] E. Doğan, "LSTM training set analysis and clustering model development for short-term traffic flow prediction," *Neural Comput. Appl.*, vol. 33, pp. 11175–11188, Jan. 2021.

HONGLING ZHENG was born in Putian, Fujian, in 2000. He is currently pursuing the B.S. degree with the College of Information Science and Engineering, Lanzhou University. He has participated in several international competitions as the Director. His research interests include traffic flow prediction and natural language processing. He was responsible for experimental design and paper writing.

XIANG LI received the master's degree from Shanghai Jiaotong University, in 2022. He works at China Electronics Technology Group Corporation. His research interests include biological simulation communication, tactical communication, and intelligent networks. He was responsible for experimental design.

YONGFENG LI is currently pursuing the Ph.D. degree with Xi'an Jiaotong University. His research interests include artificial intelligence in the field of water conservancy and flood control and public safety. He took in-charge of the data processing.

ZIQIN YAN received the bachelor's and master's degrees in computer science from California State University, East Bay. He is currently a Research and Development Engineer at The Silk Road Infoport Company Ltd. His research interests include machine learning, data analysis, and high performance distributed parallel computation. He helped design the experiment.

TINGHONG LI received the master's degree from the University of Science and Technology of China, in 2011. He is currently working at China Mobile Communications Corporation Ltd. His research was published on International Conference on Smart City as a part of 2021 IEEE Hyper-Intelligence Congress. His research interests include deep learning, reinforcement learning, and intelligent transport systems. He contributed to the revision of the format design.