Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 7586154, 11 pages
https://doi.org/10.1155/2020/7586154

Research Article

A Novel Traffic Flow Forecasting Method Based on RNN-GCN and BRB

Hailong Zhu,[1] Yawen Xie ,[1] Wei He ,[1,2] Chao Sun,[1] Kaili Zhu,[1] Guohui Zhou,[1] and Ning Ma[1]

[1] *School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China*
[2] *Rocket Force University of Engineering, Xi'an 710025, China*

Correspondence should be addressed to Wei He; 64282426@qq.com

Received 22 October 2019; Revised 29 August 2020; Accepted 7 October 2020; Published 19 October 2020

Academic Editor: Sara Moridpour

As an important part of a smart city, intelligent transport can effectively reduce energy consumption and environmental pollution. Traffic flow forecasting provides a reliable traffic dispatch basis for intelligent transport, and most of the existing prediction methods only predict a single saturation or speed and do not use the saturation and speed in a unified way. This paper proposes a new traffic flow prediction method based on RNN-GCN and BRB. First, the belief rule base (BRB) is used for data fusion to obtain new traffic flow data, then the recurrent neural network (RNN) and graph convolution neural network (GCN) model is used to obtain the time correlation of the traffic data, and finally, the traffic flow is predicted by the topology graph. The experimental results show that the method has a better performance than ARIMA, LSTM, and GCN.

1. Introduction

Urban traffic congestion not only wastes people's time and pollutes the environment but also easily leads to traffic accidents. Therefore, it is important to predict traffic flow effectively, which is a key step to solve the problems. It means that traffic flow prediction is an important part of an intelligent traffic information system. The change in road traffic flow is a real-time, nonlinear, nonstationary random process. The shorter the sampling interval, the stronger the randomness and uncertainty of traffic flow changes, which makes it difficult to accurately analyze the laws of changes and algebraic expressions. Therefore, the study of the traffic flow forecast has important practical significance.

In order to predict traffic flow, many researchers have proposed various prediction models. Generally speaking, the existing models can be divided into two categories: prediction models based on classical statistics and data-driven prediction models. The former mainly uses limited data to analyze the traffic flow data, regress and optimize the parameters of the traffic flow data, and realize the fitting prediction of the traffic data. Kumar and Vanajakshi [1] combined short-term traffic flow forecasting with seasonal ARIMA models and used limited input data to solve the applicability of the ARIMA model in traffic flow forecasting. In 2017, Kumar [2] proposed another forecasting model based on the Kalman filter; it overcomes the shortcomings that the SARIMA model is relying on a large amount of data for development. Li et al. [3] established a deep feature fusion model and used heterogeneous data to predict the average speed of time and space. Fei et al. [4] proposed a dynamic linear model (DLM) based on Bayesian inference to predict online short-term travel time on highways. Feng et al. [5] proposed a short-term traffic flow prediction algorithm based on an adaptive multikernel support vector machine (AMSVM) with spatial-temporal correlation.

The second one is a data-driven predictive model, which uses intelligent calculation methods to mine the evolution trend of historical traffic flow data to predict the future trend of the data. When using deep learning to build models, convolutional neural networks are usually used to predict traffic flow. Li et al. [6] applied a multimodal deep learning model to the estimation of missing values in heterogeneous traffic data. Li et al. [7] used an advanced multiobjective particle swarm optimization algorithm to optimize certain parameters in the deep belief network to achieve traffic flow prediction for the next day. Zhang et al. [8] proposed an ST-ResNet deep learning inverse method to predict the number of mobile populations in each region using the residual convolutional neural network. Wu et al. [9] combined CNN and LSTM to obtain spatial dependence using one-dimensional CNN and designed a short-term convolutional neural network forecasting model. Traditional convolutional neural networks can effectively extract local features of data, but they can only be applied to standard grid data. We are dealing with nongrid traffic data points, and graph convolution can directly implement convolution operations on graph-structured data. Currently, the main graph convolution methods include spatial methods and spectral methods. The spatial method directly applies the convolution kernel to the nodes on the graph and its neighborhood. The core of this method is how to select the neighborhood of the node [10, 11]. The spectral method generalizes the convolution operation on the grid data to the graph structure data through the graph Laplacian matrix [12, 13].

Although people have made great progress in traffic flow prediction using graph convolution, there are still the following problems: First, these models are for a single traffic flow data, and multiple traffic flow factors are not considered for integration. Due to the complexity and nonlinear correlation of traffic scenarios, multiple indicators of traffic flow are closely related; this paper considers them as a new traffic factor in the dataset to be used for prediction. Secondly, traffic flow is predicted at a particular period $(T + 1)$ in the future. Through the traffic factor at time T, the historical traffic flow of (T_n) and T is not considered to predict traffic flow; that is, the time correlation is not considered, which makes the accuracy of traffic flow forecasting somewhat reduced.

To the above problems, this paper gives the following solutions. First, this paper adopts a new data fusion method, and the information fusion method based on the BRB expert system is adopted for the first time. In this method, multiple indicators are combined to obtain a new dataset of traffic flow. The distribution characteristics of the output result are obtained, and the clustering analysis is performed on the output result of the sample data. Yang et al. [14] proposed a BRB system based on the evidence reasoning (ER) algorithm in 2006. In the BRB, every possible outcome of the rule is associated with the degree of belief. Such a belief rule can capture more complex and continuous causal relationships between different factors, and the traditional IF-THEN rule is its special case. BRB can be used to capture nonlinear causality as well as continuity, incompleteness, and ambiguity. It has been widely used in many fields, such as safety assessment [15], fault diagnosis [16], risk analysis [17], health assessment of engineering systems [18], fault prediction [19], system behavior prediction [20], network security prediction [21], medical and medical evaluation [22], and identification of uncertain nonlinear system [23]. Second, in this paper, a new traffic flow forecasting model based on a recurrent neural network and graph a convolutional neural network (RNN-GCN) is designed for the first time. GCN is used to implement convolution operations directly on the data of the graph structure. The topological relationship is established in any data in the normed space, the structural features of the graph are captured, and the spatial features are effectively extracted for learning. Seo et al. [24] proposed a graph convolutional recurrent network (GCRN), but it is difficult to determine the optimal combination of recurrent network and graph convolution under certain settings. Cui et al. [25] proposed a novel deep learning framework, namely, the flow graph convolutional long short-term memory neural network (TGC-LSTM), which solves the time-varying traffic patterns and the complex space constraints on the road network. Zhou et al. [26] provide a detailed review of existing graph neural network models and systematically categorize the applications. Subsequently, Yu et al. [27] proposed a GCN network of control mechanisms and applied it to traffic volume forecasting problems. RNN contains a feedback input compared to previous neurons. Guo et al. [28] propose a novel attention-based spatial-temporal graph convolutional network (ASTGCN) model to solve the traffic flow forecasting problem. The single neuron of the RNN is similar to a series of weight-sharing feed-forward neurons. After the connection, the input and output will change at the same time as the traditional neurons. But the difference is that the "historical information" of neurons at the moment on the RNN will be connected to the neurons at the next moment by weights. In this way, the RNN completes the mapping of the input and output at time t and refers to the influence of all input data on the network before t, forming a feedback network structure. Therefore, the RNN can process the sequence data; that is, the RNN will consider the information that appears before processing the current information. To sum up, the RNN-GCN traffic flow forecasting model proposed in this paper has fully considered the traffic flow status of the current and historical time nodes and established the traffic topology map to facilitate feature extraction and learning. The model is feasible and accurate, and it has been verified that the accuracy of traffic flow forecasting is improved.

In summary, a new traffic flow forecasting method based on RNN-GCN and BRB was designed for the first time. The complexity of traffic scenarios and the time correlation of traffic flow was fully considered, and the traffic flow forecasting problem was effectively solved. The paper is organized as follows: In Section 2, a feasibility analysis of the traffic flow forecast is presented, including the fusion process of the traffic flow data and the prediction of the traffic flow under the RNN-GCN model. In Section 3, a model for traffic information fusion based on BRB is given. In Section 4, a prediction model based on RNN-GCN is established. The model's effectiveness is validated by experiments and case studies in Section 5. Conclusion and future work are reported at the end of this paper.

2. Description of the Problem

Traffic flow is predicted as shown in Figure 1. First, in order to better describe the traffic congestion problem, multiple traffic flow indicators are merged to obtain new traffic indicators. Second, in order to predict the traffic flow more

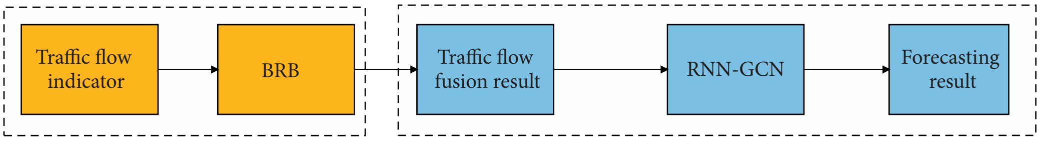

Figure 1: Traffic flow forecasting implementation process.

accurately, a new traffic flow prediction model is constructed. Finally, the predicted result is output.

2.1. Traffic Flow Information Fusion Process. In traffic flow forecasting, the datasets used in the past were only for a single indicator. Due to the complexity of the traffic scene, the forecast results are different from the real values. To this end, BRB technology is adopted in this paper, multiple indicators are merged, a nonlinear relationship between input and output is established, and a new traffic flow indicator is acquired. The BRB fusion process is shown in Figure 2.

In this process, two hypotheses are defined to describe the information fusion of traffic flow:

(1) $y(t)$ is defined as the output of the information fusion, which can be expressed as

$$y(t) = f(X(t), \alpha). \tag{1}$$

(2) $X(t)$ is defined as the input set of the information fusion and can be expressed as

$$X(t) = \{x_1(t), \ldots, x_N(t)\}, \tag{2}$$

where $x_n(t)\,(n = 1, \ldots, N)$ represents the value of the nth traffic flow indicator at time t. $f(\cdot)$ denotes the BRB-based nonlinear transformation. α denotes the parameter set of the BRB.

2.2. RNN-GCN Traffic Flow Forecasting Process. RNN-GCN is defined to describe the problem of traffic flow forecasting, and the forecast process of traffic flow is shown in Figure 3.

In this process, two hypotheses are defined to describe the traffic flow forecasting:

(1) $y(t+1)$ is defined as the forecasting result of traffic flow and can be expressed as

$$y(t+1) = g(Y(t), \beta). \tag{3}$$

(2) $Y(t)$ is defined as the input of traffic flow forecasting and can be expressed as

$$Y(t) = \{y(t-n) \cdots y(t)\}, \tag{4}$$

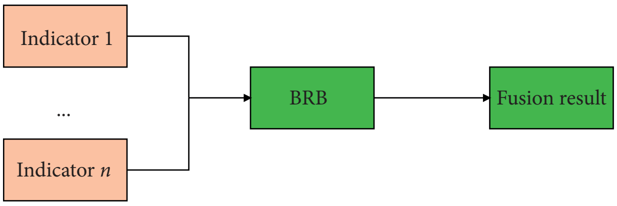

Figure 2: BRB fusion process.

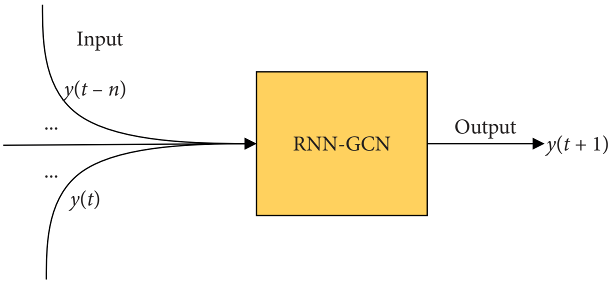

Figure 3: Traffic flow forecasting process.

where $y(t-T)$ is the traffic flow at time T. $g(\cdot)$ denotes the RNN-GCN-based nonlinear transformation. β denotes a parameter set of the RNN-GCN.

3. BRB-Based Traffic Flow Fusion Model

3.1. The Basic Structure of the Model. The BRB is composed of a series of belief rules. Initially, the belief rules are constructed. In a rule, each result has its belief degree; hence, all the results of a rule can be presented in a belief distribution. Each rule is set with a rule weight, which is defined as its relative importance to other rules. The kth rule of the BRB is shown as follows:

$$\begin{aligned}
&R_k\text{: if } x_1(t) \text{ is } A_1^k \wedge x_2(t) \text{ is } A_2^k, \ldots, \wedge x_N(t) \text{ is } A_N^k,\\
&\text{Then}\left\{\left(D_1, \beta_{1,k}\right), \ldots, \left(D_M, \beta_{M,k}\right)\right\},\\
&\text{With a rule weight } \theta_k,\\
&\text{and attribute weights } \delta_1, \delta_2, \ldots, \delta_N.
\end{aligned} \tag{5}$$

For (5), $R_k\,(k = 1, \ldots, L)$ is the kth rule of the BRB. L is defined as the number of rules in the BRB. $x_1(t)\,(i = 1, \ldots, N)$ is the input of the BRB. θ_k denotes the rule weight of the kth rule, reflecting the relative importance of kth rule. A_i^k is the reference value of the ith antecedent attribute. $A_i = \left\{A_{i,j}, j = 1, \ldots, J_i\right\}$ represents the set consisting of J_i reference values. $D_j\,(j = 1, \ldots, M)$ is the evaluation level of the output. $\beta_{j,k}$ denotes the belief degree of D_j; if $\sum_{j=1}^{N} \beta_{j,k} = 1$, then the kth rule is complete; otherwise, it is incomplete. $\delta_n (n = 1, \ldots, N)$ is the attribute weight of the kth rule, which represents the importance of the antecedent attribute.

3.2. Model Reasoning Process. The belief rule is used by BRB to realize the storage and expression of knowledge, and then the ER algorithm is used to inductive reasoning of existing knowledge. The reasoning process can be summarized as the following process:

Step 1. The rule matching calculation for the input sample is described as

$$a_i^k = \begin{cases} \dfrac{A_i^{l+1} - x_i(t)}{A_i^{l+1} - A_i^l}, & k = l\left(A_i^l \le a_i(t) \le A_i^{l+1}\right), \\ \\ \dfrac{x_i(t) - A_i^l}{A_i^{l+1} - A_i^l}, & k = l + 1, \\ \\ 0, & k = 1, 2, \ldots, K\,(k \ne l, l + 1), \end{cases} \tag{6}$$

where a_i^k is the matching degree of the input information with the kth rule; $x_i(t)$ denotes the ith antecedent attribute value in the input data; A_i^l and A_i^{l+1} are defined as the reference value of the neighboring state of the ith antecedent attribute; K is the number of belief rules.

Step 2. The weight of the matching degree and the belief rule is used to calculate the activation weight of the belief rule, which is described as

$$w_k = \theta_k \frac{\prod_{i=1}^{M} \left(a_i^k\right)^{\delta_i}}{\sum_{l=1}^{K} \theta_l \prod_{i=1}^{M} \left(a_i^l\right)^{\delta_i}}, \tag{7}$$

where w_k is the activation weight of the kth rule. θ_k denotes the rule weight of the kth rule. δ_i is represented as the ith antecedent attribute weight. M is the number of antecedent attributes.

Step 3. The ER iterative algorithm is used to combine rules to generate belief levels for different output levels.

Step 4. Using the activation weight of the belief rule, the belief degree is converted to the basic probability mass, which is described as

$$\begin{aligned} m_{m,k} &= w_k \beta_{m,k}, \\ m_{D,k} &= 1 - w_k \sum_{m=1}^{M} \beta_{m,k}, \\ \overline{m}_{D,k} &= 1 - w_k, \\ \widetilde{m}_{D,k} &= w_k \left(1 - \sum_{m=1}^{M} \beta_{m,k}\right), \end{aligned} \tag{8}$$

where $\beta_{m.k}$ is the belief degree that the kth rule produces D_m evaluation result. $m_{m.k}$ denotes the probability mass of the kth rule to produce D_m evaluation result. $m_{D,k}$ denotes the unallocated basic probability mass in the result set D. $\overline{m}_{D,k}$ is the degree of the insignificance of the kth rule. $\widetilde{m}_{D,k}$ is the degree of incompleteness of the kth belief rule. M is the set number of evaluation results.

Step 5. Let $m_{m,I(1)} = m_{m,1}, m_{D,I(1)} = m_{D,1}$ using the Dempster criterion to combine the top k rules, which are described as

$$\begin{aligned} m_{m,I(k+1)} &= L_{I(k+1)} \left[m_{m,I(k)} m_{m,k+1} + m_{m,I(k)} m_{D,k+1} + m_{D,I(k)} m_{m,k+1}\right], \\ m_{D,I(k)} &= \overline{m}_{D,I(k)} + \widetilde{m}_{D,I(k)}, \\ \widetilde{m}_{D,I(k+1)} &= L_{I(k+1)} \left[\widetilde{m}_{D,I(k)} \widetilde{m}_{D,k+1} + \widetilde{m}_{D,I(k)} \overline{m}_{D,k+1} + \overline{m}_{D,I(k)} \widetilde{m}_{D,k+1}\right], \\ \overline{m}_{D,I(k+1)} &= L_{I(k+1)} \left[\overline{m}_{D,I(k)} \overline{m}_{D,k+1}\right], \\ L_{I(k+1)} &= \frac{1}{1 - \sum_{m=1}^{M} \sum_{\substack{q = 1 \\ q \ne m}}^{M} m_{m,I(k)} m_{q,k+1}}, \quad k = 1, 2, \ldots, K, \end{aligned} \tag{9}$$

where $m_{m,I(k+1)}$ is the basic probability mass of the evaluation result D_m after combining the k rules.

Step 6. The results of information fusion are calculated.

Step 7. The belief degree in calculating the evaluation results is described as

$$\beta_m = \frac{m_{m,I(K)}}{1 - \overline{m}_{D,I(K)}}, \quad m = 1, 2, \ldots, M, \tag{10}$$

where β_m denotes the belief degree of the evaluation result D_m.

Step 8. Utility calculation is used and fusion results are generated. This process can be described as

$$y(t) = \sum_{m=1}^{M} U(D_m)\beta_m, \tag{11}$$

where $U(\cdot)$ denotes the utility formula.

In summary, the reasoning process is shown in Figure 4.

The ER algorithm uses the normalized idea when calculating the activation weight of the rule based on the input information. Compared to the traditional D-S evidence combination method, the system's ability to deal with conflict evidence is enhanced. In addition, the computational complexity of the D-S evidence combination method is an N-P difficult problem. The rule combination method of the ER algorithm is linear, and the computational complexity of the rule combination is greatly reduced. The BRB synthesizes the activated rules through the ER algorithm. The nonlinear relationship between the input and the output is established, and the feedback mechanism of the rule base is established, which realizes the possibility of self-learning.

4. Traffic Flow Forecasting Model Based on RNN-GCN

4.1. Model Structure. One of the key issues in traffic flow forecasting is the correlation of acquisition time, which is generally obtained using RNN. RNN is a special neural network structure in which the output of temporally contiguous states can be described with memory functions. In this paper, a new RNN-GCN model is proposed. The model structure is shown in Figure 5.

It is using the nodes between the hidden layers of the RNN to have connectivity. The input of the hidden layer contains not only the input of the input layer but also the output of the hidden layer at the previous moment. It mainly consists of an input layer, a hidden layer, and an output layer. The data is updated cyclically in the hidden layer. This is the way to implement the time memory function.

The hierarchical expansion of the hidden layer is shown in Figure 6. The memory function of the sample can be described as

$$S_t = f\left(W^{*}S_{t-1} + U^{*}y(t)\right), \tag{12}$$

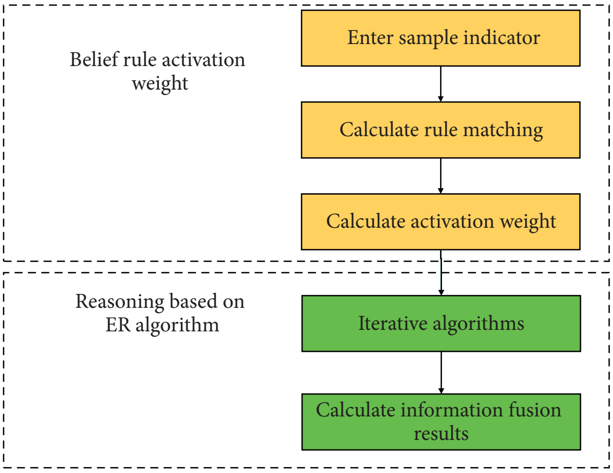
Belief rule activation weight
Enter sample indicator
Calculate rule matching
Calculate activation weight
Reasoning based on ER algorithm
Iterative algorithms
Calculate information fusion results

Figure 4: Model reasoning process.

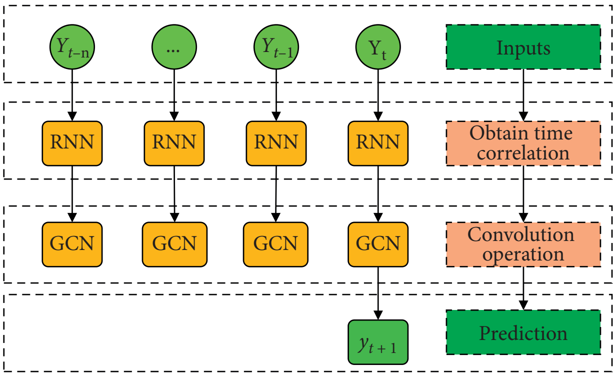

Figure 5: RNN-GCN model.

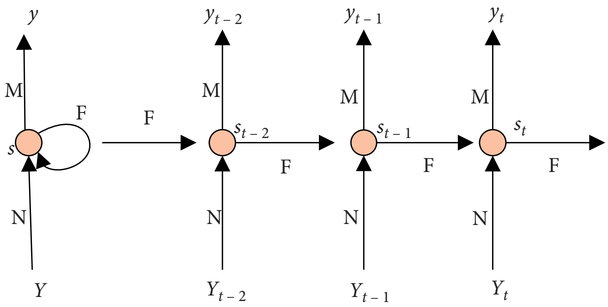

Figure 6: Hidden layer hierarchy expansion diagram.

where S_t denotes a memory of the sample at t. $y(t)$ is an input sample. W is the weight of the input. U is the weight of the $(y_{t-2}, y_{t-1}, y_t)^T$.

Through the RNN model, a piece of historical time series fragment $[y_{t-n}, y_{t-n+1}, \ldots, y_{t-1}, y_t]$ directly adjacent to the forecasting period is obtained. The convergence and divergence of vehicles on the road are gradually taking shape. Therefore, the traffic of the next moment of a node will be affected by the traffic at the previous moment.

The traffic flow forecasting of the road network is aimed at the graph structure. The structure of the graph is considered as irregular infinite-dimensional data, and the surrounding structure of each node is considered unique, so it has no translation invariance. The surrounding structure of each node may be considered unique. The data of this structure prevents traditional CNN and RNN from obtaining good prediction results. To solve the irregularity of the graph structure, GCN is introduced to learn by effectively extracting spatial features. The structure of GCN is shown in Figure 7.

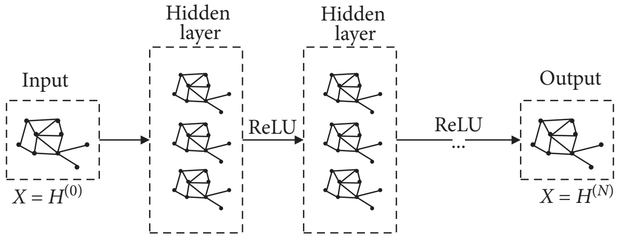

Figure 7: GCN structure.

GCN is considered as a kind of neural network dealing with the graph data structure. The core idea of GCN is considered to be to aggregate node information by using edge information, so as to generate new node representations. The essence of GCN is used to extract the spatial features of the topological graph. For graph data, node information and structural information must be considered. So GCN can automatically learn not only the characteristics of node characteristics but also the associated information between nodes.

Through the RNN model, a piece of historical time series fragment $[y_{t-n}, y_{t-n+1}, \ldots, y_{t-1}, y_t]$ directly adjacent to the forecasting period is obtained. When processing the graph structure, the spectrum analysis method [13] is considered. The graph structure of the spectrogram method is analyzed mainly by transforming the graph into an algebraic form. In this paper, the connectivity and mutual influence between nodes in the graph structure are mainly concerned. In the spectrogram method, a graph can be represented by its corresponding Laplacian matrix (L), and the properties of the graph structure can be obtained by analyzing L and its eigenvalues. The Laplacian matrix of the graph is obtained, which can be shown as

$$L = D - A. \tag{13}$$

The canonical form is described as

$$L = I_N - D^{(1/2)} A D^{(1/2)} \in \mathbb{R}^{N*N}, \tag{14}$$

where A is the adjacency matrix. I_N is the unit matrix. The degree matrix $D \in \mathbb{R}^{N\times N}$ is a diagonal matrix composed of node degrees.

The Laplacian matrix and its eigenvalues are analyzed, and the properties of the graph structure are obtained. Then, the graph signal $y = y_t^f \in \mathbb{R}^N$ is transformed through Fourier, and the result is expressed as $\widehat{y} = U^T y$. Graph convolution is a kind of convolution operation that is implemented by defining a diagonal linear operator in the Fourier domain to equivalently replace the classic convolution operator [29]. The specific operation is defined as a convolution operation on the graph with the convolution kernel g_θ:

$$g_\theta^* y = g_\theta(L) y = g_\theta\left(U \Lambda U^T\right) y = U g_\theta(\Lambda) U^T y, \tag{15}$$

where $\wedge = \operatorname{diag}([\lambda_0, \ldots, \lambda_{N-1}]) \in \mathbb{R}^{N\times N}$ is a diagonal matrix composed of the eigenvalues of L. U is the Fourier basis [27].

But when the scale of the graph is large, it is expensive to decompose the Laplacian matrix directly. Therefore, the approximate expansion of Chebyshev polynomials is used to solve the problem:

$$\begin{aligned} g_\theta \times y &= g_\theta(L) y \approx \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{L}) y, \\ \widetilde{L} &= \frac{2}{\lambda_{\max}} L - I_N, \\ T_k(y) &= 2y T_{k-1}(y) - T_{K-2}(y), \end{aligned} \tag{16}$$

where $\theta_k \in \mathbb{R}^k$ is the coefficient of Chebyshev polynomial. $\lambda_{\max}$ is the largest eigenvalue of L. $T_k(y)$ is the recursive definition of Chebyshev polynomial.

To sum up, the GCN can be used to convolute the undirected graph, and the traffic flow data can be topologically processed. The GCN model is trained and learned to predict the traffic flow, which solves the problem that the traditional neural network cannot predict the complex traffic network in real time and accurately.

4.2. Model Implementation Process. According to the complex characteristics of the traffic scene, the RNN-GCN traffic forecasting model is constructed, and the specific implementation process is shown in Figure 8. The detailed steps are described as follows:

(1) Obtain the time correlation: through the RNN model, historical time series are constructed, and continuous output in time is obtained
(2) Serialize the graph structure: the dataset is converted into a form of a topological graph structure as a data input to the GCN model
(3) Train the model: the training data is input, a series of convolution operations are operated, and the model is trained
(4) Predict the traffic flow: the trained model is used to predict traffic flow

5. Case Analysis

In this section, actual datasets (SZ-taxi) are selected to realize data fusion and traffic flow forecasting. Without loss of generality, traffic speed and traffic saturation are used as traffic information in the experimental part. SZ-taxi is a dataset of taxi trajectories in Shenzhen from January 1 to January 31, 2015. 156 main roads are selected in Luohu District as the research area. The experimental data mainly

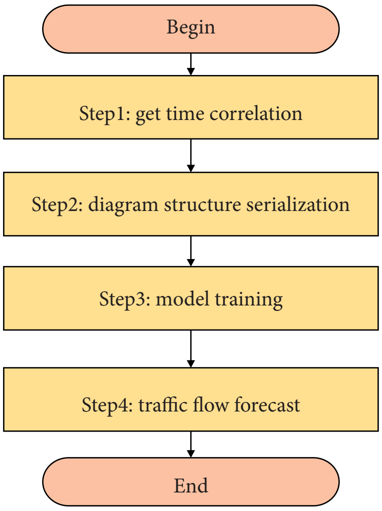
Begin
Step1: get time correlation
Step2: diagram structure serialization
Step3: model training
Step4: traffic flow forecast
End

Figure 8: RNN-GCN model implementation process.

includes two parts: one is a 156 ∗ 156 adjacency matrix, which describes the spatial relationship between roads, each row represents a road, and the value in the matrix represents the connectivity between roads. The other is the feature matrix, which describes the change of the speed on each road over time. Each row represents the traffic speed on the road at different times, and each column is a road. The traffic speed and traffic saturation are calculated on each road every 15 minutes. Traffic speed is the calculated average speed of all cars on a certain road. Traffic saturation is the ratio of the current number of road vehicles to the current street carrying capacity.

This case includes two parts. Firstly, BRB is used to fuse the two indicators of traffic speed and traffic saturation to get a traffic flow fusion result. Secondly, the results of BRB fusion are put into RNN-GCN for prediction. Finally, the prediction results are obtained. Figure 9 shows the traffic flow prediction process of the experiment.

5.1. BRB Data Fusion Case.

In the case of BRB data fusion, traffic saturation and traffic speed were combined to obtain a new dataset of traffic flow. The output result of the sample data is analyzed, the distribution characteristics of the output result are acquired, and the system rule number and the result evaluation level related parameter are determined.

To express domain knowledge in BRB, reference points and reference values need to be given. In this BRB model, two indicators of traffic speed (x_1) and traffic saturation (x_2) are selected as the input of the BRB system, and y is the output to reflect traffic conditions. The selection of reference points and reference values is based on expert knowledge.

There are 8 reference points for x_1, namely, zero (Z), very small (VS), small (S), a little small (LS), middle (M), a little large (LL), large (L), and very large (VL). There are 6 reference points for x_2, namely, zero (Z), small (S), a little small (LS), middle (M), large (L), and very large (VL).

There are 4 reference points for y, namely, poor (P), middle (M), good (G), and very good (VG); namely, $D = \{D_1, D_2, D_3, D_4\} = \{\text{P, M, G, VG}\}$. The four values of D correspond to the probability of P, M, G, and VG, respectively (Tables 1–3).

According to the expert system, a belief rule table as shown in Table 4 is constructed, and there are 48 belief rules.

By inputting the traffic data samples, the matching degree calculation is performed. The belief rule table is used. The traffic saturation is merged with traffic speed. And the nonlinear relationship between the input and the output is established. And a new traffic flow forecasting index is obtained.

5.2. RNN-GCN Traffic Flow Forecasting Case

5.2.1. Prove the Performance of the Model.

Through the analysis of experimental results and comparative test analysis, the traffic information at the next moment can be predicted based on the traffic data at the historical time point. This section evaluates the prediction performance of the RNN-GCN model compared to other prediction models. Without loss of generality, only the traffic speed is used as the traffic information for the experimental part.

We compare our model with the following three baselines:

(i) ARIMA: autoregressive integrated moving average method is a well-known time series analysis method for predicting future values.

(ii) GCN: graph convolutional neural network uses graph convolution operation to process topological graph structure data.

(iii) LSTM: long short-term memory network is a special RNN model.

Figure 10 is a comparison diagram of traffic speed prediction results and real values in a day. It can be seen from the figure that the forecasting result of the model in the day is close to the true value of the data. It shows that the RNN-GCN model can obtain the best forecasting performance through training. Figure 10 demonstrates the effectiveness of the RNN-GCN model.

5.2.2. Define Evaluation Indicators.

To effectively evaluate the performance of RNN-GCN, three evaluation indicators are used to measure the prediction effect of the model, where y_t represents the actual traffic flow data at time t and $\hat{y}_t$ represents the output of traffic flow forecasting by the model. NUM is the total number of test samples, including the following:

(i) Mean absolute error (MAE):

$$\text{MAE} = \frac{1}{\text{NUM}} \sum_{i=1}^{\text{NUM}} \left| y_t - \hat{y}_t \right|. \tag{17}$$

MAE can well reflect the actual situation of predicted value error and is the first choice for model

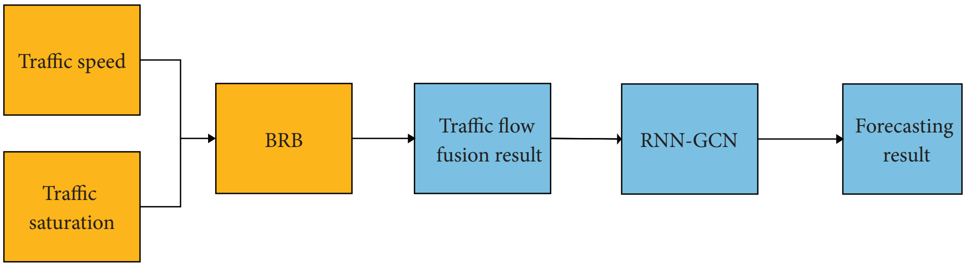

Figure 9: Traffic flow prediction process.

Table 1: Reference points and reference values for x_1.

Reference point	Reference value
Z	0
VS	10
S	20
LS	30
M	40
LL	50
L	60
VL	70

Table 2: Reference points and reference values for x_2.

Reference point	Reference value
Z	0
S	20
LS	40
M	60
L	80
VL	100

Table 3: Reference points and reference values for y.

Reference point	Reference value
P	2
M	4
G	6
VG	8

prediction effect evaluation: the smaller the value is, the better the prediction effect is.

(ii) Root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{\text{NUM}} \sum_{i=1}^{\text{NUM}} \left(y_t - \widehat{y}_t\right)^2}. \quad (18)$$

RMSE is used to measure the deviation between the predicted value and the true value and is more sensitive to outliers in the data: the smaller the value is, the better the prediction effect is.

(iii) Accuracy (ACC):

Table 4: Belief rule table.

Serial number	Rule weight	x_1	x_2	$\{D_1, D_2, D_3, D_4\}$
1	1	Z	VL	{0, 0, 0, 1}
2	1	Z	L	{0, 0, 0.2, 0.8}
3	1	Z	M	{0, 0.1, 0.3, 0.6}
4	1	Z	LS	{0.1, 0.5, 0.2, 0.2}
5	1	Z	S	{0.8, 0.2, 0, 0 }
6	1	Z	Z	{1, 0, 0, 0}
7	1	VS	VL	{0, 0, 0, 1}
8	1	VS	L	{0, 0, 0.35, 0.65}
9	1	VS	M	{0.1, 0.2, 0.5, 0.2}
10	1	VS	LS	{0.15, 0.25, 0.45, 0.15}
11	1	VS	S	{0.7, 0.2, 0.1, 0 }
12	1	VS	Z	{1, 0, 0, 0 }
13	1	S	VL	{0, 0, 0, 1}
14	1	S	L	{0, 0, 0.4, 0.6}
15	1	S	M	{0.15, 0.25, 4, 0.1}
16	1	S	LS	{0.3, 0.4, 0.2, 0.1}
17	1	S	S	{0.5, 0.3, 0.1, 0.1}
18	1	S	Z	{1, 0, 0, 0}
19	1	LS	VL	{0, 0, 0, 1}
20	1	LS	L	{0.3, 0.3, 0.2, 0.2}
21	1	LS	M	{0.35, 0.3, 0.25, 0.1}
22	1	LS	LS	{0.4, 0.2, 0.35, 0.15}
23	1	LS	S	{0.5, 0.35, 0.1, 0.05}
24	1	LS	Z	{1, 0, 0, 0}
25	1	M	VL	{0, 0, 0, 1}
26	1	M	L	{0.2, 0.3, 0.3, 0.2 }
27	1	M	M	{0.35, 0.3, 0.3, 0.05}
28	1	M	LS	{0.35, 0.45, 0.2, 0}
29	1	M	S	{0.45, 0.4, 0.15, 0}
30	1	M	Z	{1, 0, 0, 0}
31	1	LL	VL	{0, 0, 0, 1 }
32	1	LL	L	{0.2, 0.3, 0.2, 0.3}
33	1	LL	M	{0.35, 0, 0.35, 0.3}
34	1	LL	LS	{0.45, 0.3, 0, 0.25}
35	1	LL	S	{0.55, 0.2, 0.25, 0}
36	1	LL	Z	{1, 0, 0, 0 }
37	1	L	VL	{0, 0, 0, 1 }
38	1	L	L	{0.3, 0.35, 0.25, 0.1 }
39	1	L	M	{0.45, 0.4, 0, 0.15}
40	1	L	LS	{0.45, 0.4, 0.15, 0}
41	1	L	S	{0.65, 0.3, 0, 0.05 }
42	1	L	Z	{1, 0, 0, 0}
43	1	VL	VL	{0, 0, 0, 1}
44	1	VL	L	{0.35, 0.4, 0.2, 0.05}
45	1	VL	M	{0.55, 0.45, 0, 0}
46	1	VL	LS	{0.75, 0.2, 0.05, 0}
47	1	VL	S	{0.9, 0.1, 0, 0}
48	1	VL	Z	{1, 0, 0, 0 }

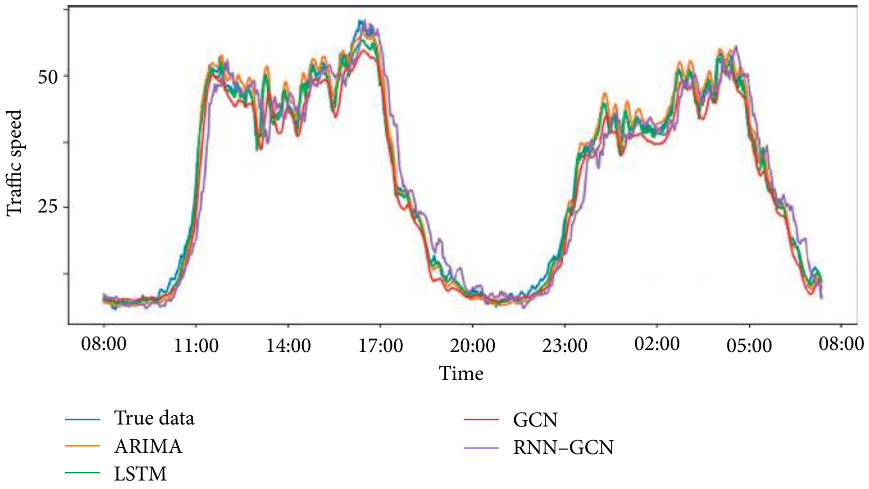

Figure 10: Comparison of traffic speed forecasting experiments.

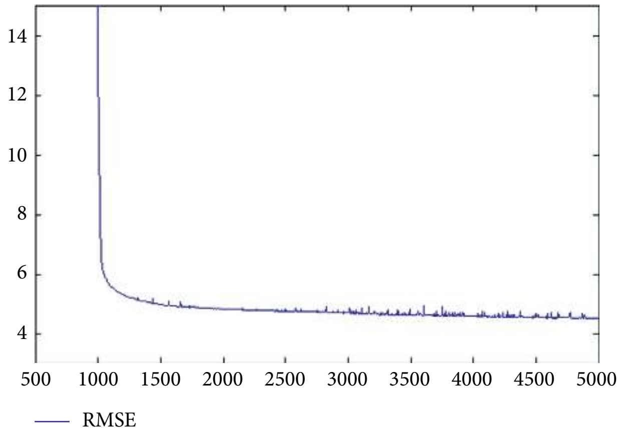

Figure 11: RNN-GCN RMSE diagram.

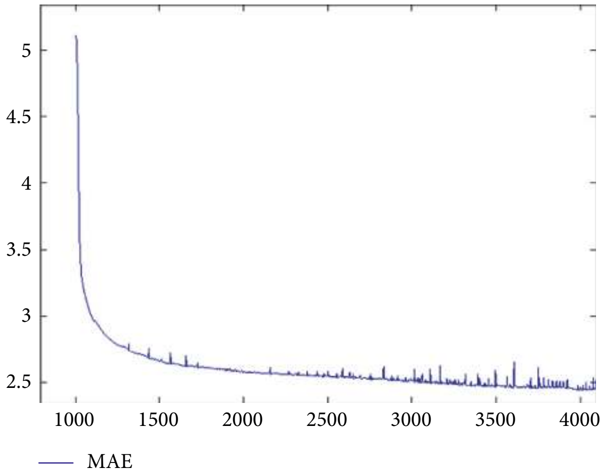

Figure 12: RNN-GCN MAE diagram.

$$\mathrm{ACC} = 1 - \frac{\|y - \hat{y}\|_F}{\|y\|_F}. \tag{19}$$

ACC is used to detect the prediction precision: the larger the value is, the better the prediction effect is. $\|\cdot\|_F$ represents the norm of this matrix which is Frobenius norm, y is a matrix of true values, and $\hat{y}$ is a matrix of predicted values.

5.3. Analysis of Experimental Results. Traffic indicators are fused by BRB, and a dataset of traffic flow is generated. In the experiment of traffic flow forecasting, 80% of the dataset is used for model training and 20% of the dataset is used for model testing. Figures 11–13 show the decrease of RAME and MAS and the increase of ACC with the increase of training times. This shows the effectiveness of the model when the BRB fusion data is input into RNN-GCN.

We compare our models with the three baseline methods. Table 5 shows the average results of traffic flow prediction performance. It can be seen from Table 5 that our RNN-GCN achieves the best performance in the same datasets in terms of all evaluation metrics. We can observe that the prediction results of the traditional time series analysis methods are usually not ideal, demonstrating those methods' limited abilities to model nonlinear and complex traffic data.

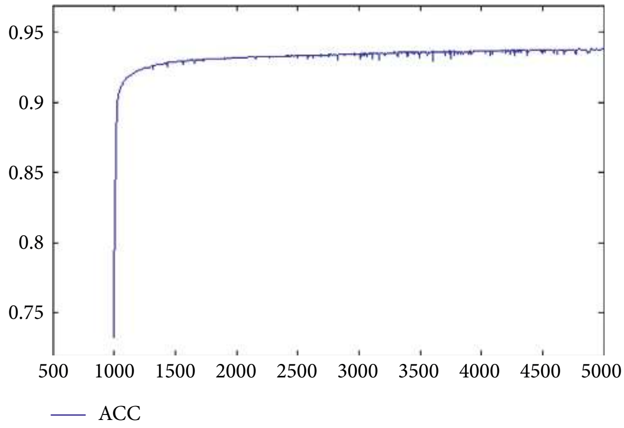

Figure 13: RNN-GCN ACC diagram.

Table 5: Performance comparison of different methods.

Model	RMSE	MAE	ACC
ARIMA	18.2123	16.2192	0.4282
LSTM	20.3218	19.2910	0.7859
GCN	15.4120	13.7009	0.8907
RNN-GCN	14.8279	13.4857	0.9063

6. Conclusions

In this paper, a new traffic flow forecasting method based on RNN-GCN and BRB is proposed. First, the BRB model is used for data fusion, and new traffic flow data is obtained; second, the RNN model is used to obtain time correlation; finally, the GCN model is used for traffic forecasting. Through experiments, the method proposed in this paper is better than other forecasting models; the accuracy has improved. But obtaining temporal correlation through the time attention mechanism is not considered. Therefore, our next step will be to optimize the predictive model by adding strategies such as time attention mechanisms and further improve the predictive ability of the model.

Data Availability

In the study of this paper, the dataset used is a public dataset, and the access address is https://github.com/lehaifeng/T-GCN/tree/master/data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Heilongjiang Province of China under Grant no. F2018023, the Ph.D. Research Start-Up Foundation of Harbin Normal University under Grant no. XKB201905, and the Natural Science Foundation of School of Computer Science and Information Engineering, Harbin Normal University, under Grant nos. JKYKYY2020002 and JKYKYY202010.

References

[1] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal,” *European Transport Research Review*, vol. 7, no. 3, 2015.

[2] S. V. Kumar, “Traffic flow prediction using Kalman filtering technique,” *Procedia Engineering*, vol. 187, pp. 582–587, 2017.

[3] L. Li, X. Qu, J. Zhang, Y. Wang, and B. Ran, “Traffic speed prediction for intelligent transportation system based on a deep feature fusion model,” *Journal of Intelligent Transportation Systems*, vol. 23, no. 6, pp. 605–616, 2019.

[4] X. Fei, C.-C. Lu, and K. Liu, “A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1306–1318, 2011.

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

[6] L. Li, B. Du, Y. Wang, L. Qin, and H. Tan, “Estimation of missing values in heterogeneous traffic data: application of multimodal deep learning model,” *Knowledge-Based Systems*, vol. 194, Article ID 105592, 2020.

[7] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, “Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm,” *Knowledge-Based Systems*, vol. 172, pp. 1–14, 2019.

[8] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, “Predicting citywide crowd flows using deep spatio-temporal residual networks,” *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[9] C.-H. Wu, J.-M. Ho, and D. T. Lee, “Travel-time prediction with support vector regression,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[10] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in *Proceedings of the International Conference on Machine Learning*, pp. 2014–2023, New York City, NY, USA, June 2016.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: data-driven traffic forecasting,” 2017, https://arxiv.org/abs/1707.01926.

[12] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” 2013, https://arxiv.org/abs/1312.6203.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” 2016, https://arxiv.org/abs/1606.09375.

[14] J. Yang, J. Liu, J. Wang, H. S. Sii, and H. Wang, “Belief rule-base inference methodology using the evidential reasoning approach-RIMER,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 36, no. 2, pp. 266–285, 2006.

[15] G. Li, Z. Zhou, C. Hu, L. Chang, Z. Zhou, and F. Zhao, “A new safety assessment model for complex system based on the conditional generalized minimum variance and the belief rule base,” *Safety Science*, vol. 93, pp. 108–120, 2017.

[16] Z. Feng, Z. Zhou, C. Hu, X. Yin, G. Hu, and F. Zhao, “Fault diagnosis based on belief rule base with considering attribute correlation,” *IEEE Access*, vol. 6, pp. 2055–2067, 2017.

[17] G. Kong, D.-L. Xu, R. Body, J.-B. Yang, K. Mackway-Jones, and S. Carley, “A belief rule-based decision support system for clinical risk assessment of cardiac chest pain,” *European Journal of Operational Research*, vol. 219, no. 3, pp. 564–573, 2012.

[18] X. Yin, B. Zhang, Z. Zhou, X. Han, Z. Wang, and G. Hu, “A new health estimation model for CNC machine tool based on infinite irrelevance and belief rule base,” *Microelectronics Reliability*, vol. 84, pp. 187–196, 2018.

[19] Z.-J. Zhou, C.-H. Hu, D.-L. Xu, M.-Y. Chen, and D.-H. Zhou, “A model for real-time failure prognosis based on hidden Markov model and belief rule base,” *European Journal of Operational Research*, vol. 207, no. 1, pp. 269–283, 2010.

[20] C. Hu, C. Hu, D. Xu, J. Yang, and D. Zhou, “New model for system behavior prediction based on belief rule based systems,” *Information Sciences*, vol. 180, no. 24, pp. 4834–4864, 2010.

[21] G.-Y. Hu, Z.-J. Zhou, B.-C. Zhang, X.-J. Yin, Z. Gao, and Z.-G. Zhou, “A method for predicting the network security situation based on hidden BRB model and revised CMA-ES algorithm,” *Applied Soft Computing*, vol. 48, pp. 404–418, 2016.

[22] M. S. Hossain, F. Ahmed, F. T. Johora, and K. Andersson, “A belief rule based expert system to assess tuberculosis under uncertainty,” *Journal of Medical Systems*, vol. 41, no. 3, p. 43, 2017.

[23] Y.-W. Chen, J.-B. Yang, C.-C. Pan, D.-L. Xu, and Z.-J. Zhou, “Identification of uncertain nonlinear systems: constructing belief rule-based models,” *Knowledge-Based Systems*, vol. 73, pp. 124–133, 2015.

[24] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, “Structured sequence modeling with graph convolutional recurrent networks,” in *Proceedings of the International Conference on Neural Information Processing*, pp. 362–373, Siem Reap, Cambodia, December 2018.

[25] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” 2019, https://arxiv.org/abs/1802.07007.

[26] J. Zhou, G. Cui, Z. Zhang et al., “Graph neural networks: a review of methods and applications,” 2018, https://arxiv.org/abs/1812.08434.

[27] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting,” 2017, https://arxiv.org/abs/1709.04875.

[28] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 922–929, 2019.

[29] M. Henaff, J. Bruna, and Y. LeCun, “Deep convolutional networks on graph-structured data,” 2015, https://arxiv.org/abs/1506.05163.