**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

# Traffic Flow Predication by Using Graph Convelution Network

Saja A Fadhil[1,2] and Ping Lou[1*]

[1]school of information engineering wuhan university of technology, wuhan, hubei 430070, china

[2]school of electronic engineering, college of engineering, university of diyala, iraq

*correspondence: louping@whut.edu.cn , saja.9494@yahoo.com

**ARTICLE INFO**

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

**ABSTRACT**

These days, a wide range of applications and electrical gadgets employ machine learning techniques to solve issues and forecast future conditions. Since there are so many cars on the road and traffic congestion is readily caused by them, one of the problems that most large cities face is traffic jams. Although traffic congestion are bad for the environment, their effects may be reduced with careful planning. Because it may prevent traffic congestion by using its predictive knowledge, one of the most intriguing topics for intelligent transportation systems is traffic prediction. It is quite difficult for academics to develop or use a model that will function properly in various conditions when it comes to traffic prediction. One of the most important aspects of traffic prediction is capturing both geographical and temporal connections. Taking use of model combinations is one approach to capturing both dependencies. the model combine three layer the graph convolution network and the long short term memory . Graph convolutional network is used by the model to learn the spatial dependency based on road network architecture, and long short term memory are used to learn the short-time trend in time series. To further increase prediction accuracy, the attention mechanism was included to modify the weight assigned to various time points and compile global temporal information.

**Keywords:** traffic forecasting, temporal graph convolutional network, spatial dependence, temporal dependence.

## 1-INTRODUCTION

Traffic forecasting is a crucial element of traffic control, transportation planning and management, and intelligent transportation systems [12, 15–17] . Complex spatiotemporal relationships have made accurate real-time traffic forecasting extremely difficult. Periodicity and propensity are signs of temporal dependency, which is the idea that the status of the traffic varies over time. Due to the transfer of upstream traffic state to downstream portions and the retroactive effects of downstream traffic state on the upstream section, changes in traffic state are subject to the structural topology of road networks, a phenomenon known as spatial dependence[10] .

Therefore, achieving the traffic forecasting assignment requires taking into account the topological properties of the road network as well as its complicated temporal features. There are two types of traffic forecasting models now in use: parametric and non-parametric. Historical averages, time series [1, 14], linear regression models [27], and Kalman filtering models [23] are examples of common parametric models. Traditional parametric models rely on the stationary hypothesis even if they make use of straightforward techniques. These models are unable to account for the nonlinearity and unpredictability of traffic conditions or to mitigate the impact of unforeseen circumstances like traffic accidents. When given sufficient historical data, non-parametric models may automatically learn the statistical laws of data, which enables them to effectively address these issues. K-nearest [2], support vector regression (SVR) [11, 30], fuzzy logic [34], Bayesian network [28], and neural network models are examples of common non-parametric models.

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Because deep learning is developing so quickly, deep neural network models have recently drawn a lot of interest from academics [22, 26]. Because they can employ self-circulation mechanism and simulate temporal dependency, (RNNs), long short-term memory (LSTM) [13], and gated recurrent units (GRUs) [7] have been effectively used in traffic forecasting [20, 25]. Nevertheless, these models ignore spatial dependency and solely take into account the temporal change of traffic status. Convolutional neural networks (CNNs) have been incorporated by several researchers into their models in order to effectively quantify spatial dependency.

Wu et al. [31] combined an LSTM with a CNN to create a feature fusion framework for short-term traffic flow forecasting. Using two LSTMs, the framework investigated the short-term fluctuations and periodicity of traffic flow while capturing the spatial properties of the flow through a one-dimensional CNN. Cao and associates. [6] presented an end-to-end model known as ITRCN, which used a CNN to collect network flows and convert interactive network traffic to pictures.

GRU was also used by ITRCN to extract temporal characteristics. An investigation demonstrated that this method's predicting error was 14.3% and 13.0% greater than those of GRU and CNN, respectively. Yu et al. [36] used LSTM and DCNN, respectively, to capture temporal dynamics and spatial correlation.

Additionally, they demonstrated the advantages of SRCN through an analysis of Beijing traffic network data. Even if CNN is useful for Euclidean data[9], such images and grids, it is still limited when used to traffic networks since these networks have non-Euclidean architecture.

Graph convolutional network (GCN) has emerged in recent years [18], It has advanced quickly and is able to get over the aforementioned constraints while capturing the structural features of networks [19, 35, 37]. Furthermore, RNNs and its variations employ sequential processing across time and are better able to retain the most recent data, making them appropriate for capturing changing short-term patterns. As though The proximity of time alone cannot discern the relative value of distinct time points. For this reason, above design the model for traffic forecasting. the model combine of the three layer of GCN and connected to LSTM.

## 2-RELATED WORK

It has been demonstrated that intelligent transport systems (ITS) maximize traffic flow, safety, and overall transportation efficiency. Precise traffic flow forecasting is one of the issues that ITS faces [29].

Among the methods that have been investigated to address this traffic flow forecasting issue, statistical models are a frequently employed option. Utilizing past traffic data, the statistical models identify patterns and trends that may be applied to forecasting.

The autoregressive integrated moving average (ARIMA) statistical model is predicated on the idea that there exists a robust pattern in the historical data pertaining to traffic [14]. Due to the dynamic nature of urban areas, traffic patterns constantly change over time and diverge from past traffic norms. Traditional machine learning approaches are also employed to anticipate traffic flow, and ML models have an advantage than statistic models in that they can handle more complicated patterns and considerably bigger data sets for analysis. Support vector regression (SVR) [26] and k-nearest neighbors (KNN) [27] are the popular models.

Machine learning models may take into account variables like the time of day, the local site of interest, and the weather [30]. Since deep learning has advanced, deep learning models have outperformed standard machine learning [13]. Two deep learning models that are often utilized in traffic flow forecasting are the gated recurrent unit (GRU) and long-short term memory (LSTM) [26]. However, the issue with all of the aforementioned models is that they ignore the spatial characteristics of traffic data, which prevents them from having a strong understanding of the nearby roadways. In addition to the

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

previously discussed methods, typical machine learning and deep learning models are unable to handle graph-structured data because neighboring roads have an impact on the following road [21].

## 3-METHODOLOGY LSTM3GCN MODEL

3.1- Definition of problems

In this study, traffic forecasting is done on urban highways to estimate future traffic conditions based on past traffic conditions. Traffic flow, speed, and density are all considered to be part of the traffic condition. The only definition of "traffic state" in this study is "traffic speed."

Definition 1 of a road network G: The set of road sections is denoted by G = (V , E) ,where V = ($v_1$,$v_2$, ..... , $v_N$), and the number of road sections is indicated by N. This represents the topological structure of the urban road network. The edges, or set E, represent the links between the various portions of the route. The adjacent matrix A € $R^{N,N}$ contains all of the connectivity data. Road sections index the rows and columns, and each entry's value represents the connection between relate road d The whole connectivity data is kept in the adjacent matrix A € $R^{N,N}$, where road sections serve as the index for rows and columns. Each entry's value represents the connection between related road sections. In the event that there isn't a road link, the entry value is 0 (an unweighted graph) or non-negative (a weighted graph).

Definition 2. by calculating a sequence representation from its adjacency matrix, enhances the traffic flow forecast. In particular, traffic prediction can benefit from the application of self-attention to represent the temporal interdependence of traffic patterns over extended periods of time. In traffic prediction jobs, this is useful since historical traffic patterns have a significant influence on future traffic circumstances. With the application of a self-attention mechanism, deep learning models are able to assess the relative relevance of various input sequence segments [48]. Each input sequence must have three sets of vectors computed for the self-attention mechanism to function: query, key, and value vectors. The proper amount of attention that the model should assign is determined by the weights given to each element in the sequence. In this approach, we use a scaled dot-product self-attention with the linear transformation of the adjacency matrix and compute the attention-weighted feature using Figure as follows: The adjacency matrix A is divided into three copies: Query Q, Key K, and Value V, the similarity between the Q and each K can be used to calculate the attention score, which reflects the importance of different temporal moments. The attention score is normalized by softmax, and finally, multiply the value V by the normalized Q and K to get the attention-weighted feature Â

$$\hat{A} = softmax\left(\frac{Q\,k^{(t)}}{\sqrt{kd}}\right)v \text{ .....(1)}$$

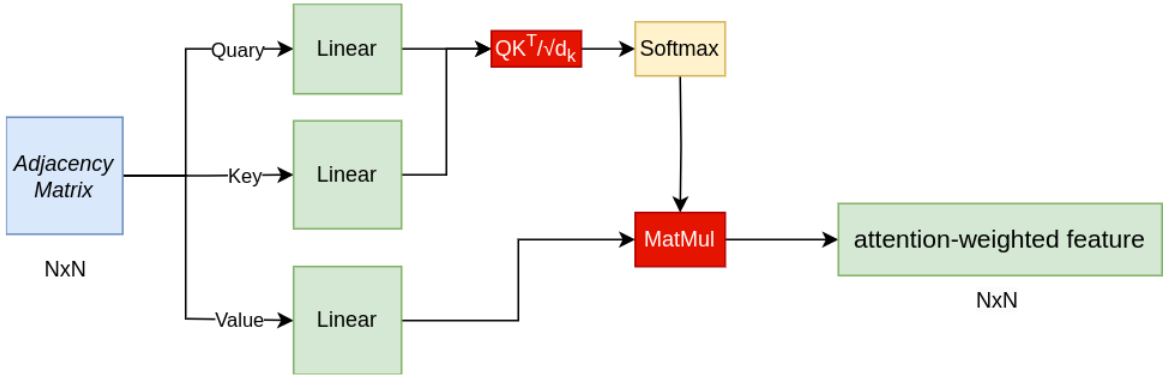


Fig -1 the Architecture of the adjacent matrix .

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The weighted sum of the vectors for each node in a graph, denoted as Â, is what will be passed to the GCN layers in place of the adjacency matrix Â. The model is given X as traffic input (traffic speed or flow) and A as its adjacency matrix. This forces the GCN layer to extract node relationships from the adjacency matrix Â rather than taking them directly to LSTM , and the model receives the architecture.

### 3.2 Graph neural network (GNN)

Graph Neural Networks (GNNs): GNNs are a kind of neural network that works with graphs, which are mathematical structures made up of nodes and edges that represent entities and relationships between them. GNNs can be used to learn graph representations and perform different tasks like node classification, link prediction, and graph classification [51]. GNNs work by gathering information from neighboring nodes and updating each node's representation based on this information [23]. Traditionally, traffic flow forecasting has been accomplished through the use of statistical models, time series analysis, and machine learning techniques like Support Vector Regression (SVR). Nevertheless, the geographical dependencies and correlations among traffic data, which are crucial components for precise traffic flow prediction, are not well captured by these systems. GNNs are able to grasp the intricate correlations between many data sources, including traffic flow, meteorological information, and road network layout. By representing network topology as a graph and identifying the geographical connections between traffic data, GNNs can overcome these drawbacks [45].

The aggregator and updater are two crucial parts of a Graph Neural Network (GNN) that collaborate to update each node's hidden state depending on the hidden states of its neighbors [45].

After receiving the hidden states of nearby nodes, the aggregator function combines or processes these states to create a node representation. Subsequently, the concealed state of the central node is updated using this node representation. A variety of functions, such as mean, max, or sum pooling, as well as more intricate ones like attention mechanisms or graph convolution, can be used as the aggregate function. In [50] As an illustration, consider a mean pooling aggregator, which creates a node representation by averaging the hidden states of nearby nodes.

$$a_i^{(l)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} h_j^{(l-1)} \qquad \text{.....(2)}$$

where |N(i)| is the number of neighboring nodes of node i and $a(l)_i$ is the node representation of the hidden states of neighboring nodes for node i at layer l [45]. The updater function comes after the aggregator function, taking in the node representation and the node's previous hidden state to produce the new hidden state for the node [50].

### 3.3 Graph Convolutional Network (GCN)

Graph structure processing may be done using semi-supervised models called GCNs. They represent a graph fields version of CNNs. GCNs have made significant advancements in a wide range of applications, including unsupervised learning [18] , document classification [9], and picture classification [5]. In GCNs, the spatial domain and spectrum convolutions are included in the convolutional mode . In this investigation, the former was used. Compared to a single-layer GCN, the multi-layer GCN improves the model's ability to capture more intricate representations. The model's ability to comprehend the influences of various nodes' neighbors is improved by adding layers. Excessive layers have been shown to improve a model's performance in sparse graphs. This is due to the fact that in both dense and sparse graphs, the GCN model is unable to collect enough environmental data in one layer. This means that adding additional layers makes it possible to record a wider range of environmental data . The product of figure filter $g_\theta(L)$, which is built in the Fourier domain, and signal x on the graph is known as spectrum convolution. $g_\theta(L) * x = U g_\theta (U^T x)$, where _ is

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

a model parameter, L is the graph Laplacian matrix, U is the eigenvector of normalized Laplacian matrix $L = I_N - D^{-1/2} A D^{-1/2} = U\lambda U^T$ , and $U^T x$ is the graph Fourier transformation of x. x can also be promoted to $X \in R^{N\times N}$. In order to capture the spatial characteristics of a graph, GCNs can perform the spectrum convolutional operation, taking into account the graph node and first-order adjacent domains of nodes. This allows GCNs to replace the convolutional operation in anterior CNNs, given the characteristic matrix X and adjacent matrix A. Moreover, many networks are superposed using the hierarchical propagation rule. A multilayer GCN model has the following expression:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}\theta^{(l)}\right) \quad \text{.....3}$$

where $\tilde{A} = A + I_N$ is an adjacent matrix with self connection structures, $I_N$ is an identity matrix, D is a degree matrix, $D_{ii} = \sum \tilde{A}_{ij}$ , $H(l) \in R^{N\times l}$ is the output of layer l, $\theta^{(l)}$ is the parameter of layer l, and σ(.) is an activation function used for nonlinear modeling.Generally at three layer of GCN . We define the GCN as the following:

$$Z = f(X,A) = softmax\left(\hat{A}\,Re\,L\,U\left(\hat{A}XW^{(0)}\right)W^{(1)} + W^{(2)}\right) \quad \text{.....(4)}$$

In other words, this study learned spatial dependence through the GCN model [18]. GCNs can encode the topological structures of road networks and the attributes of road sections simultaneously by determining the topological relationship between the central road section and the surrounding road sections.

### 3.4- long short –term memory (LSTM )

The LSTM model is an effective recurrent neural system that was created specifically to solve the exploding/vanishing gradient issues that frequently occur while learning long-term dependencies, even in cases where the minimal time delays are extremely lengthy[40]. Generally, a constant error carousel (CEC), which keeps the error signal inside each unit's cell, can stop this. Since the input and output gates, which together create the memory cell, are added to the CEC, these cells are actually recurrent networks in and of themselves. This makes for an intriguing design. One time step lag is shown by the self-recurrent connections, which represent feedback. An input gate, an output gate, a forget gate, and a cell make up a vanilla LSTM unit. suggested the forget gate as a way to enable the network to reset its state, even though it was not originally a component of the LSTM network. The three gates control the information flow related to the cell, and the cell retains data for random periods of time. Since this is the most often used LSTM architecture, LSTM will be referred to as the vanilla variant in the next sections. That being said, this does not mean that it is always the better option. The memory blocks that make up the LSTM architecture are a collection of sub-networks that are connected recurrently. The purpose of the memory block is to control the flow of information through non-linear gating units and preserve its state across time. The gates, input signal x(t), output signal y(t), activation functions, and peephole connections make up the architecture of a vanilla LSTM block, as shown in Figure 2. All of the gates and the block's input are repeatedly linked to the block's output.

Let's imagine a network with N processing blocks and M inputs in order to better understand how the LSTM model functions. This recurrent neural system's forward pass is explained below.

**Block input**: This phase is all about updating the block input component, which is made up of the output of that LSTM unit ($y^{(t-1)}$ )from the previous iteration and the current input $x^{(t)}$ . [39] This may be done as follows:

$$z(t) = g(W_z x^{(t)} + R_z y^{(t-1)} + b_z) \quad \text{.....(5)}$$

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

where $W_z$ and $R_z$ are the weights associated with $x^{(t)}$ and $y^{(t-1)}$ , respectively, while bz stands for the bias weight vector.

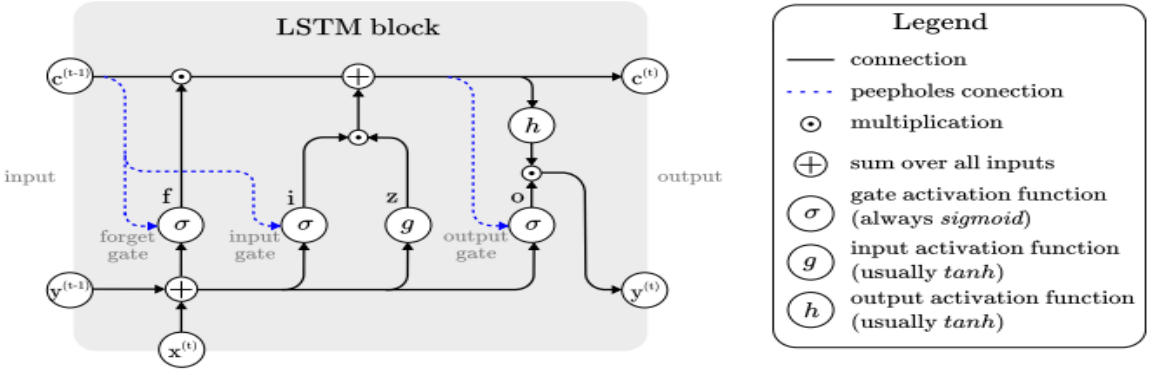


Fig -2 the Architecture of LSTM Block.

**Input gate** : it combines the cell value $c^{(t-1)}$ from the previous iteration, the output of that LSTM unit, and the current input $x^{(t)}$ is updated in this step. The process is shown by the following equation:

$$i^{(t)} = \sigma (W_i x^{(t)} + R_i y^{(t-1)} + p_i \odot c^{(t-1)} + b_i) \text{.....( 6 )}$$

where Wi, Ri, and pi are the weights associated with $x^{(t)}$, $y^{(t-1)}$, and $c^{(t-1)}$, respectively, and bi stands for the bias vector associated with this component. Since $\odot$marks the point-wise multiplication of two vectors.

The LSTM layer decides which data should be kept in the network's cell states $c^{(t)}$ in the earlier phases[38]. This involved choosing the input gate activation values, $i^{(t)}$, and candidate values, $z^{(t)}$ , that might be added to the cell states.

**Forget gate** : The LSTM unit chooses which data from its earlier cell states $c^{(t-1)}$ should be eliminated in this stage. Thus, the current input $x^{(t)}$, the outputs $y^{(t-1)}$ and the state $c^{(t-1)}$ of the memory cells at the previous time step (t − 1), the peephole connections, and the bias terms $b_f$ of the forget gates are used to compute the activation values $f^{(t)}$ of the forget gates at time step t. Here is how this may be accomplished:

$$f^{(t)} = \sigma (W_f x^{(t)} + R_f y^{(t-1)} + p_f \odot c^{(t-1)} + b_f) \text{ .....(7)}$$

The weights associated with $x^{(t)}$, $y^{(t-1)}$, and $c^{(t-1)}$ are represented by $W_f$, $R_f$, and $p_f$, respectively, while $b_f$ stands for the bias weight vector. Cell This step computes the cell value by combining the values of the block input $z^{(t)}$, the input gate $i^{(t)}$, and the forget gate $f^{(t)}$ with the previous cell value. This can be done as shown below[40] .

$$c^{(t)} = z^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)} \text{ .....(8)}$$

**Output gate** : This phase calculates the output gate, which combines the current input $x^{(t)}$ , the output of that LSTM unit $y^{(t-1)}$ and the cell value $c^{(t-1)}$ in the last iteration. This may be done as indicated below:

$$o^{(t)} = \sigma (W_o x^{(t)} + R_o y^{(t-1)} + p_o \odot c^{(t)} + b_o) \text{.....(9)}$$

where Wo, Ro and po are the weights associated with x(t) , y(t−1) and c(t−1) , respectively, while bo denotes for the bias weight vector.

**Block output** Finally, we calculate the block output, which combines the current cell value c(t) with the current output gate value as follows:

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

$$y^{(t)} = g(c^{(t)}) \odot o^{(t)} \text{ .....(10)}$$

In the above steps, σ, g and h denote point-wise non-linear activation functions. The logistic sigmoid σ (x) = (1/1+e$^{1-x}$) is used as a gate activation function, while the hyperbolic tangent g(x) = h(x) = tanh(x) is often used as the block input and output activation function. Although vanilla LSTM already performs very well, several works studied the possibilities to improve performance. For example, developed the Extended LSTM model, further improving the accuracy of predictions in several application fields by enhancing the memory capability. This suggests that theoretical improvements can still be made to an already state-of-the-art performing architecture. It seems appropriate to mention that the functionality of this architecture inspired the authors in to enhance the training of very deep networks. The gating mechanism was employed in the so-called highway networks to allow an unimpeded information flow across many layers. This could be considered another proof-of-concept, showing that the gates work. The process of looking for ways to make the model better was already underway. In order to maximize the sequence learning capabilities of LSTM, the authors searched for an alternate architecture. Through gradient descent, they were able to evolve memory cell architectures that could learn context-sensitive formal languages; in many aspects, their performance was similar to that of LSTM. The authors created Long Short-Term Memory Spiking Neural Networks (LSNN), which include adaptive neurons, by building atop recurrent networks of spiking neurons.

It was demonstrated that the performance is extremely similar to that of LSTM during experiments when the size of LSNN was comparable to that of LSTM. This is just one more example of how precise LSTM is and always will be.

### 3.5 **LSTM3GCN model**

The model built for combine three layer of GCN connected to one LSTM , encapsulating geographical and temporal relationships. For both long- and short-term traffic flow forecasting, T-GCN is utilized. The concept of "spatial dependency" in traffic relates to how traffic patterns at one place in a road network might affect other areas. Because of their geographical linkages, the traffic flow across various sites in the network is interconnected. Traffic is temporally dependent, meaning that historical traffic circumstances affect current traffic situations. In addition to the current circumstances, past traffic patterns and earlier states also influence the flow of traffic at any particular time [49].

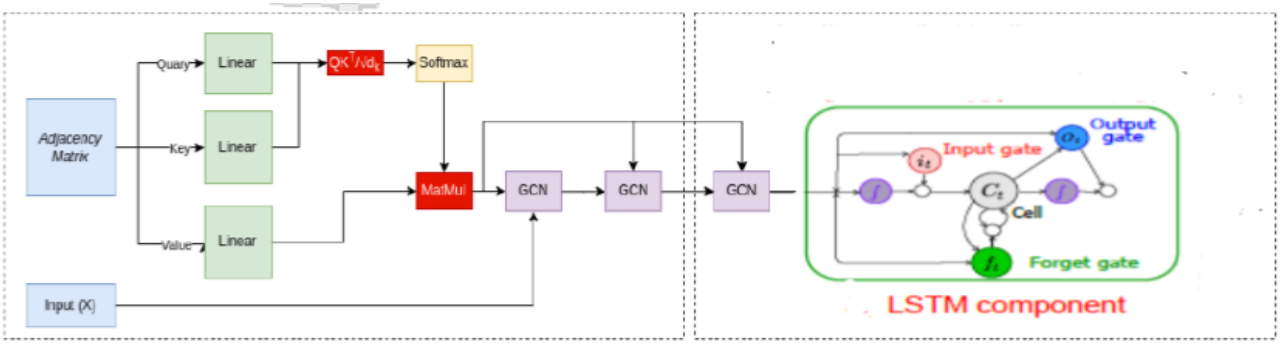


Fig -3- the architecture of the LSTM3GCN model.

A multi-layer GCN improves the model's ability to capture more intricate representations than one-layer GCN alone. The model's ability to comprehend the various nodes' neighbor effect improves with the addition of layers. It has been demonstrated that when a graph is sparse, a model performs better with more layers. This is a result of the GCN model's inability to collect enough environmental data in a single layer in thick and sparse graphs. Therefore, adding more than one layer makes it possible to record a wider range of environmental data [5]. Increasing the sensitivity of GCNs to the topology of a

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

graph that has complicated topologies may result in a performance loss. To preserve the capacity to extract both local and global information from the graph, it is crucial to maintain a balance between this sensitivity. When the graph attention mechanism is used to enhance the models' ability to handle complicated graph topologies, the sensitivity of GCNs can be resolved [42]. Three layers make up the multi-layer GCN architecture that we have developed in this work. The GCN is defined as follows:

$$Z = f(X, A) = softmax\left(\hat{A}\, Re\, L\, U\left(\hat{A}XW^{(0)}\right)W^{(1)} + W^{(2)}\right) \text{.....(11)}$$

Three layers make up the architecture, and the weight matrices are W0, W1, and W2.Given an adjacency matrix A and input X, The first GCN layer receives the ˆA and X as input. These inputs are passed through three convolutional graph layers. The second and third convolutional graph layers receive the generated adjacency matrix as input. Lastly, the final output for the model is obtained by applying the ReLU activation function to the resultant matrix .The output of the GCN layers is the input for the LSTM model, which produces a prediction and a hidden state for the subsequent time stamp.

## 4-EXPERIMENTS

4.1 Data Description

**PeMSD4:** It is collected by the Caltrans Performance Measurement System (PeMS) with a time granularity of 5 minutes. The PeMSD4 dataset refers to the traffic flow data in the San Francisco Bay Area, which contains traffic information of 307 loop detectors from 1/Jan/2018 to 28/Feb/2018.it used python language .

4.2 Evaluation Metrics

To evaluate the prediction performance of the model, the error between real traffic speed and predicted results is evaluated on the basis of the following metrics:

(1) Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2} \text{.....(12)}$$

(2) Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \text{.....(13)}$$

(3) Mean Absolute Percentage Error:

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \text{.....(14)}$$

Prediction error is specifically measured using RMSE, MAE, and MAPP. High prediction precision is shown by small RMSE, MAPE, and MASE values. Predictive precision is gauged by accuracy, with a high accuracy number being desirable.

### 4.3 Performance Comparison

We compare LSTM3GCN with several baseline models, including:

GCN: Captures spatial dependencies but lacks temporal modeling capabilities.

TGCN: Integrates GCN with temporal graph convolution but does not incorporate memory mechanisms.

ASTGCN: Uses an attention-based spatiotemporal GCN but is computationally expensive.

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

HGCN: Employs hierarchical graph convolution to improve spatial representation but lacks long-term dependency modeling.

### 4.4 Experimental result analysis and discussion

In this paper, we propose deep learning framework LSTM3GCN for traffic prediction, integrating graph convolution network and long short term memory block. Experiments show that model involved manually setting the learning rate and epoch for the datasets to 0.0001 and 100, respectively, based on prior experiences. The traffic information in the next 15, 30,45,60,90, and 120 min is predicted. The predicted results are compared with results from the other model are

GCN, TGCN, ASTGCN, and HGCN.

Table 4-1 Experimental results of all models predicting the next 15 minutes under the PeMS04 dataset.

| model | Evaluation matrices | | |
|---|---|---|---|
| | MAE | RMSE | MAPE % |
| GCN | 41.37 | 58.51 | 0.442 |
| TGCN | 209.04 | 261.46 | 0.919 |
| ASTGCN | 23.76 | 35.48 | 0.179 |
| HGCN | 38.72 | 55.33 | 0.410 |
| **LSTM3GCN** | **18.47** | **29.66** | **0.121** |

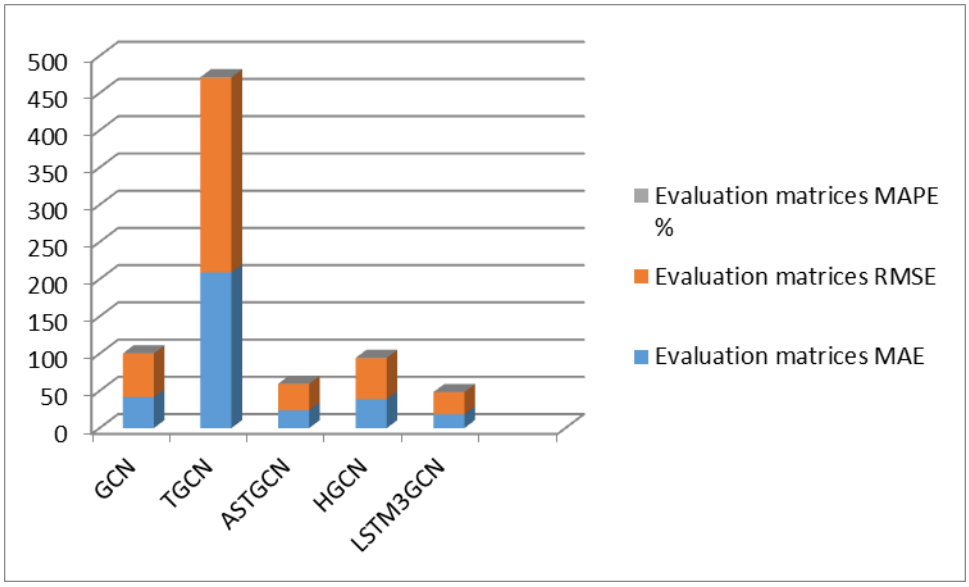


Figure 4-2 Comparison of RMSE, MAE, and MAPE values of all models in predicting the next 15 minutes.

This chapter conducts experimental verification based on the PeMSD4 dataset and the three evaluation metrics RMSE, MAE, and MAPE. Tables 4-1,present the experimental results of all models in predicting traffic flow for the next 15 minutes. The comparative models include those based on traditional time series methods and those based on deep neural network methods. From the above table, it can be intuitively seen that LSTM3GCN has superior prediction performance.

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

Table 4-2 Experimental results of all models predicting the next 30 minutes under the PeMSD4 dataset.

| model | Evaluation matrices | | |
|---|---|---|---|
| | **MAE** | **RMSE** | **MAPE %** |
| GCN | 41.91 | 57.7 | 0.451 |
| TGCN | 161.71 | 219.53 | 0.884 |
| ASTGCN | 61.87 | 80.62 | 0.937 |
| HGCN | 290.88 | 234.89 | 0.514 |
| **LSTM3GCN** | **20.51** | **32.48** | **0.135** |

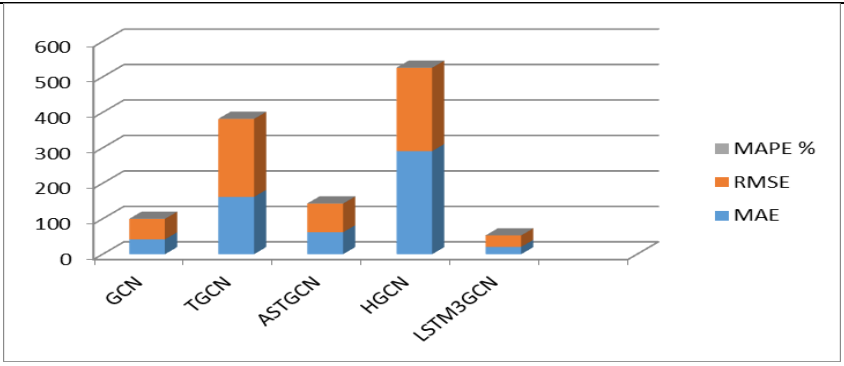


Figure 4-3 Comparison of RMSE, MAE, and MAPE values of all models in predicting the next 30 minutes.

Tables 4-2,present the experimental results of all models in predicting traffic flow for the next 30 minutes. The comparative models include those based on traditional time series methods and those based on deep neural network methods. From the above table, it can be intuitively seen that LSTM3GCN has superior prediction performance.

Table 4-3 Experimental results of all models predicting the next 45 minutes under the PeMSD4 dataset.

| model | Evaluation matrices | | |
|---|---|---|---|
| | **MAE** | **RMSE** | **MAPE %** |
| GCN | 46.77 | 64.77 | 0.567 |
| TGCN | 119.48 | 166.45 | 0.814 |
| ASTGCN | 33.11 | 48.34 | 0.3105 |
| HGCN | 180.67 | 353.68 | 0.358 |
| **LSTM3GCN** | **22.15** | **34.72** | **0.146** |

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

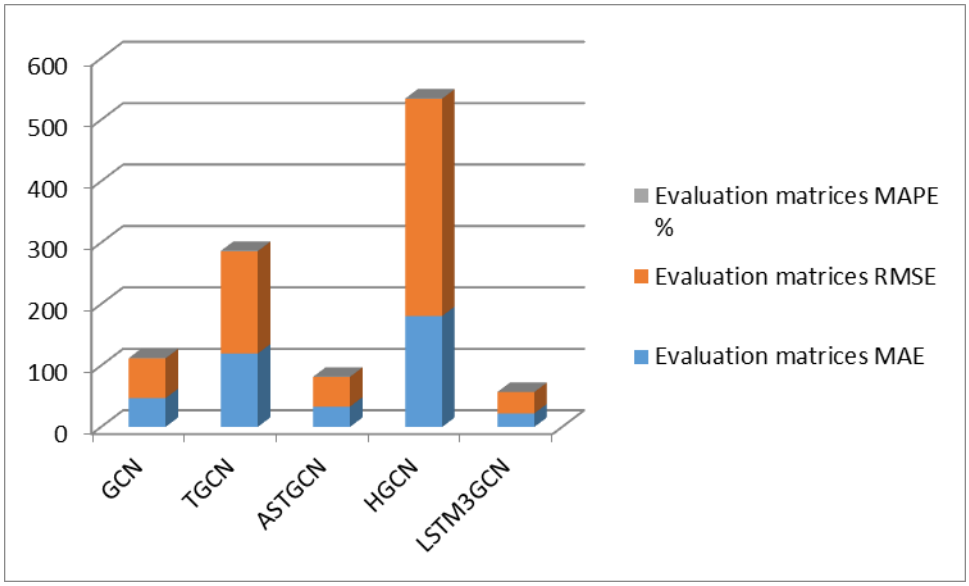


Figure 4-4 Comparison of RMSE, MAE, and MAPE values of all models in predicting the next 45 minutes.

Tables 4-3,present the experimental results of all models in predicting traffic flow for the next 45 minutes. The comparative models include those based on traditional time series methods and those based on deep neural network methods. From the above table, it can be intuitively seen that LSTM3GCN has superior prediction performance.

Table 4-4 Experimental results of all models predicting the next 60 minutes under the PeMSD4 dataset.

| **model** | **Evaluation matrices** | | |
|---|---|---|---|
| | **MAE** | **RMSE** | **MAPE %** |
| GCN | 37.56 | 54.82 | 0.323 |
| TGCN | 120.89 | 172.30 | 0.902 |
| ASTGCN | 45.54 | 63.63 | 0.503 |
| HGCN | 290.67 | 334.68 | 0.514 |
| **LSTM3GCN** | **23.93** | **37.17** | **0.158** |

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

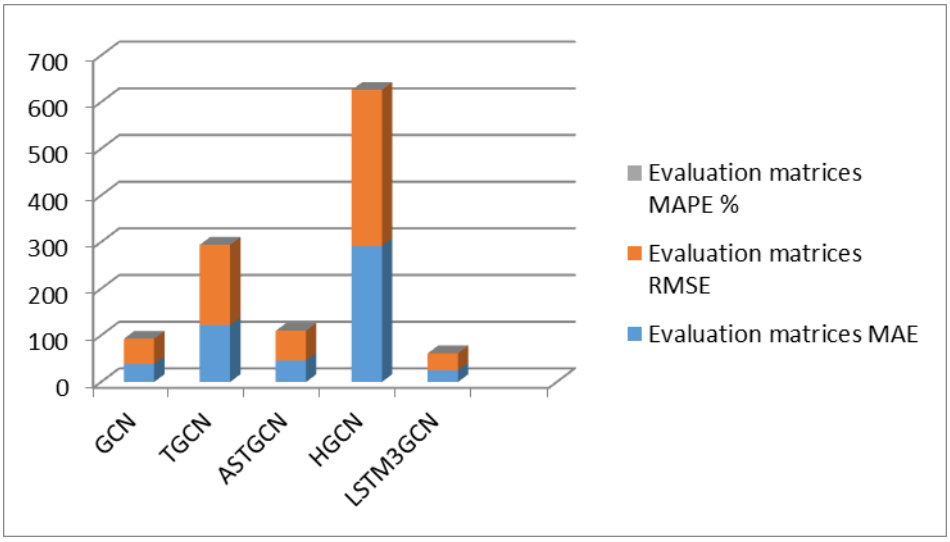


Figure 4-5 Comparison of RMSE, MAE, and MAPE values of all models in predicting the next 60 minutes.

Tables 4-4,present the experimental results of all models in predicting traffic flow for the next 60 minutes. The comparative models include those based on traditional time series methods and those based on deep neural network methods. From the above table, it can be intuitively seen that LSTM3GCN has superior prediction performance.

Table 4-5 Experimental results of all models predicting the next 90 minutes under the PeMSD4 dataset.

| **model** | **Evaluation matrices** | | |
|---|---|---|---|
| | **MAE** | **RMSE** | **MAPE %** |
| GCN | 39.80 | 58.40 | 0.369 |
| TGCN | 112.48 | 155.35 | 0.962 |
| ASTGCN | 50.54 | 70.41 | 0.546 |
| HGCN | 43.08 | 61.57 | 0.303 |
| **LSTM3GCN** | **25.79** | **39.68** | **0.171** |

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

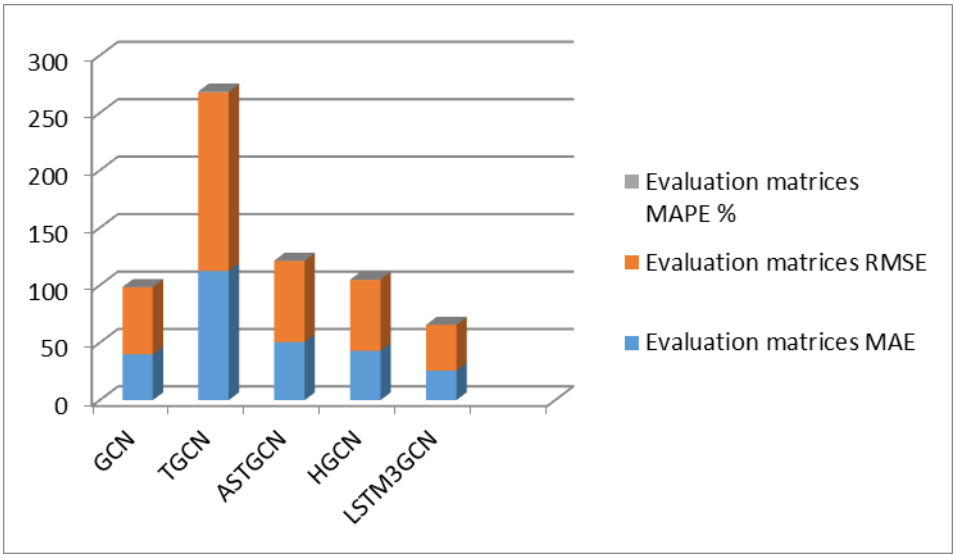


Figure 4-6 Comparison of RMSE, MAE, and MAPE values of all models in predicting the next 90 minutes.

Tables 4-5,present the experimental results of all models in predicting traffic flow for the next 90 minutes. The comparative models include those based on traditional time series methods and those based on deep neural network methods. From the above table, it can be intuitively seen that LSTM3GCN has superior prediction performance.

Table 4-6 Experimental results of all models predicting the next 120 minutes under the PeMSD4 dataset.

| **model** | **Evaluation matrices** | | |
|---|---|---|---|
| | **MAE** | **RMSE** | **MAPE %** |
| GCN | 49.65 | 70.63 | 0.504 |
| TGCN | 150.41 | 199.89 | 0.897 |
| ASTGCN | 57.94 | 79.09 | 0.679 |
| HGCN | 44.12 | 62.61 | 0.347 |
| **LSTM3GCN** | **23.59** | **36.95** | **0.156** |

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

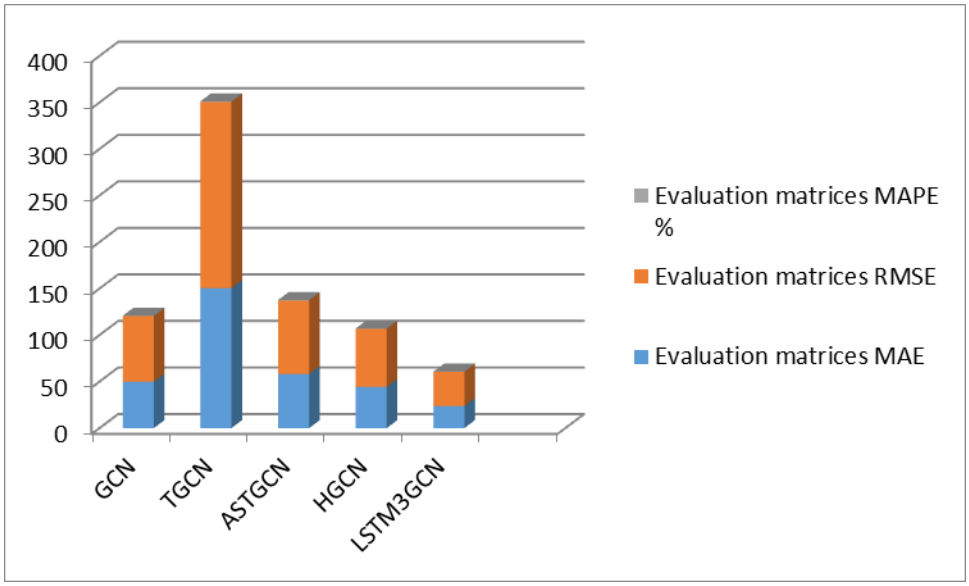


Figure 4-7 Comparison of RMSE, MAE, and MAPE values of all models in predicting the next 120 minutes.

As shown in Figure 4-2 and Table 4-1, when predicting traffic flow for the next 15 minutes, the MAE value of LSTM3GCN is reduced by 44.64%,8.83%,77.7% ,and 47.7% Compared to the GCN,TGCN,ASTGCN,HGCN models ,Respectively. In term RMSE ,it is reduced by 50.6%,11.3%, 83.5% ,and 53.6% , respectively. for the MAPE value , it is reduced by 27.3%,13.1%,67.5%, and 29.5%, compared to them, respectively.

As shown in Figure 4-3 and Table 4-2, when predicting traffic flow for the next 30 minutes, the MAE value of LSTM3GCN is reduced by 48.9%,12.6%,33.15% ,and 7.1% Compared to the GCN,TGCN,ASTGCN,HGCN models ,Respectively. In term RMSE ,it is reduced by 56.2%,14.7%, 40.28% ,and 13.8% , respectively. for the MAPE value , it is reduced by 29.9%,15.27%,14.4%, and 26.2%, compared to them, respectively.

As shown in Figure 4-4 and Table 4-3, when predicting traffic flow for the next 45 minutes, the MAE value of LSTM3GCN is reduced by 47.35%,18.53%,66.8% ,and 12.25% Compared to the GCN,TGCN,ASTGCN,HGCN models ,Respectively. In term RMSE ,it is reduced by 53.6%,20.8%, 71.8% ,and 9.8% , respectively. for the MAPE value , it is reduced by 25.7%,17.9%,47.02%, and 40.78%, compared to them, respectively.

As shown in Figure 4-5 and Table 4-4, when predicting traffic flow for the next 60 minutes, the MAE value of LSTM3GCN is reduced by 63.7%,19.7%,52.54% ,and 8.2% Compared to the GCN,TGCN,ASTGCN,HGCN models ,Respectively. In term RMSE ,it is reduced by 76.8%,21.5%, 71.8% ,and 11.1% , respectively. for the MAPE value , it is reduced by 48.9%,17.5%,31.4%, and 30.7%, compared to them, respectively.

As shown in Figure 4-6 and Table 4-5, when predicting traffic flow for the next 120 minutes, the MAE value of LSTM3GCN is reduced by 64.7%,22.92%,51.02% ,and 59.8% Compared to the GCN,TGCN,ASTGCN,HGCN models ,Respectively. In term RMSE ,it is reduced by 67.9%,25,54%, 56.3% ,and 64.4% , respectively. for the MAPE value , it is reduced by 46.34%,17.7%,31.31%, and 56.43%, compared to them, respectively.

As shown in Figure 4-7 and Table 4-6, when predicting traffic flow for the next 120 minutes, the MAE value of LSTM3GCN is reduced by 47.50%,15.6%,40.7% ,and 53.4% Compared to the GCN,TGCN,ASTGCN,HGCN models ,Respectively. In term RMSE ,it is reduced by 52.3%,18.4%,

**Journal of Information Systems Engineering and Management**
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

46.7% ,and 59.01% , respectively. for the MAPE value , it is reduced by 30.95%,17.39%,22.97%, and 44.95%, compared to them, respectively.

## 6- CONCLUSION AND FUTURE WORK

The simultaneous capture of global temporal dynamics and spatial correlations is made possible by a traffic forecasting technique called LSTM3GCN, which makes the process easier. The network of metropolitan roadways is built as a graph, with road traffic speed represented as a node's property on the graph. The suggested approach uses a three-layer GCN to capture the spatial relationships depending on the topological features of the road network. In the meanwhile, the dynamic variance. The LSTM records each successive historical traffic speed. Additionally, discovered that a typical technique is to combine GCN with LSTM in order to achieve the best possible performance and when compare the result the combine the GCN with LSTM is more better when connected with GRU. Convolution and pooling layers were employed in these hybrid models to lower the problem's dimensionality while significantly lowering the redundancy in the data. Further research into these components might result in LSTM variations with better prediction skills. the worldwide trend of temporal variation is taken in by the attention mechanism and assembled. A multi-layer GCN improves the model's ability to capture more intricate representations than one-layer GCN alone. The model's ability to comprehend the various nodes' neighbor effect improves with the addition of layers. It has been demonstrated that when a graph is sparse, a model performs better with more layers. This is a result of the GCN model's inability to collect enough environmental data in a single layer in thick and sparse graphs. Therefore, adding more than one layer makes it possible to record a wider range of environmental data [5].And the datset used , It is collected by the Caltrans Performance Measurement System (PeMS) with a time granularity of 5 minutes. The PeMSD4 dataset refers to the traffic flow data in the San Francisco Bay Area, which contains traffic information of 307 loop detectors from 1/Jan/2018 to 28/Feb/2018.

## 7-REFRENCES

[1] NTedjopurnomo, D.A.; Bao, Z.; Zheng, B.; Choudhury, F.; Qin, A.K. A survey on modern deep neural network for traffic prediction:Trends, methods and challenges. IEEE Trans. Knowl. Data Eng. 2020, 34, 1544–1561

[2] N. S. Altman. An introduction to kernel and nearestneighbor nonparametric regression. American Statistician, 46(3):175–185, 1992.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[4] . Bengio, Y., . Simard, P., and . Frasconi, P. Learning longterm dependencies with gradient descent is difficult. IEEE Trans Neural Netw, 5(2):157–166, 2002.

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun. Spectral networks and locally connected networks on graphs. 2013.

[6] Xiaofeng Cao, Yuhua Zhong, Zhou Yun, Wang Jiang, and Weiming Zhang. Interactive temporal recurrent convolution network for traffic prediction in data centers. IEEE Access, PP(99):1–1, 2017.

[7] Kyunghyun Cho, Bart Van Merrienboer, Dzmitry Bahdanau,and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. Computer Science, 2014.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[9] Michal Defferrard, Xavier Bresson, and Pierre Vandergheynst.Convolutional neural networks on graphs with fast localized spectral filtering, 2016.

[10] Chun Jiao Dong, Chun Fu Shao, Zhuge Cheng-Xiang, and Meng Meng. Spatial and temporal characteristics for congested traffic on urban expressway. Journal of Beijing University of Technology, 38(8):1242–1246+1268, 2012.

[11] Gui Fu, GuoQiang Han, Feng Lu, and ZiXin Xu. Shortterm traffic flow forecasting model based on support vector machine regression. Journal of South China University of Technology, 41(9):71–76, 2013.

[12] Lei Gao, Xingquan Liu, Yu Liu, Pu Wang, Min Deng, Qing Zhu, and Haifeng Li. Measuring road network topology vulnerability by ricci curvature. Physica A: Statistical Mechanics and its Applications, 527:121071, 2019.

[13] Alex Graves. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997.

[14] Victoria J. Hodge, Rajesh Krishnan, Jim Austin, John Polak, and Tom Jackson. Short-term prediction of traffic flow using a binary neural network. Neural Computing and Applications, 25(7-8):1639–1655, 2014.

[15] Hai Jun Huang. Dynamic modeling of urban transportation networks and analysis of its travel behaviors. Chinese Journal of Management, 2005.

[16] Yuan Jian and Bingquan Fan. Synthesis of short-term traffic flow forecasting research progress. Urban Transport of China, 2012.

[17] Liu Jing and Guan Wei. A summary of traffic flow forecasting methods. Journal of Highway and Transportation Research & Development, 2004.

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

[19] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Graph convolutional recurrent neural network: Data-driven traffic forecasting. CoRR, abs/1707.01926, 2017.

[20] J.W. C. Van Lint, S. P. Hooqendoorn, and H. J. Van Zuvlen. Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks. Transportation Research Record Journal of the Transportation Research Board, 1811(1):347369, 2002.

[21] Minh Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. Computer Science, 2015.

[22] MMorav?k,MSchmid, N Burch, V Lisy, D Morrill, N Bard, T Davis, K Waugh, M Johanson, and M Bowling. Deepstack: Expert-level artificial intelligence in heads-up nolimit

poker. Science, 356(6337):eaam6960, 2017.

[23] Iwao Okutani and Yorgos J. Stephanedes. Dynamic prediction of traffic volume through kalman filtering theory. Transportation Research Part B Methodological, 18(1):1–11, 1984.

[24] Nikolaos Pappas and Andrei Popescu-Belis. Multilingual hierarchical attention networks for document classification. 2017.

[25] Fu Rui, Zhang Zuo, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In Youth Academic Conference of Chinese Association of Automation, 2016.

[26] D Silver, J Schrittwieser, K Simonyan, I Antonoglou, A. Huang, A Guez, T Hubert, L Baker, M. Lai, and A Bolton. Mastering the game of go without human knowledge. Nature, 550(7676):354–359, 2017.

[27] Hongyu Sun, Chunming Zhang, and Bin Ran. Interval prediction for traffic time series using local linear predictor. In International IEEE Conference on Intelligent Transportation Systems, 2004.

[28] Shiliang Sun, Changshui Zhang, and Guoqiang Yu. A bayesian network approach to traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems, 7(1):124–132, 2006.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[30] Chun-Hsin Wu, Jan-Ming Ho, and D. Lee. Travel-time prediction with support vector regression. Intelligent Transportation Systems, IEEE Transactions on, 5:276 – 281, 01 2005.

[31] Yuankai Wu and Huachun Tan. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. 2016.

[32] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks, 2017.

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image

caption generation with visual attention, 2015.

[34] Hongbin Yin, S. C.Wong, Jianmin Xu, and C. K.Wong. Urban traffic flow prediction using a fuzzy-neural approach. Transportation Research Part C, 10(2):85–98, 2002.

[35] Byeonghyeop Yu, Yongjin Lee, and Keemin Sohn. Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (gcn). Transportation Research Part C: Emerging Technologies, 114:189–204, 2020.

[36] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors, 17(7):1501–, 2017.

[37] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. IEEE

Transactions on Intelligent Transportation Systems, 2019.

[38] Bellec G, Salaj D, Subramoney A, Legenstein R, Maass W (2018) Long short-term memory and learningto-learn in networks of spiking neurons. In: Advances in neural information processing systems, pp 787–797

[39] Bhunia AK, Konwer A, Bhunia AK, Bhowmick A, Roy PP, Pal U (2019) Script identification in natural scene image and video frames using an attention based convolutional-LSTM network. Pattern Recognit 85:172–184

[40] Zhang P, Ouyang W, Zhang P, Xue J, Zheng N (2019b) SR-LSTM: state refinement for LSTM towards pedestrian trajectory prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 12085–12094

[41] Shengyou Wang, Jin Zhao, Chunfu Shao, Chunjiao Dong, and Chaoying Yin. Truck traffic flow prediction based on lstm and gru methods with sampled gps data. IEEE Access, 8:208158–208169, 2020. doi: 10.1109/ACCESS.2020.3038788.

[42] Xinjue Wang, Esa Ollila, and Sergiy A. Vorobyov. Graph neural network sensitivity under probabilistic error model, 2022.

[43] Tianshu Wu, Kunqing Xie, Guojie Song, and Cheng Hu. A multiple svr approach with time lags for traffic flow prediction. In 2008 11th International IEEE Conference on Intelligent Transportation Systems, pages 228–233, 2008. doi: 10.1109/ITSC.2008. 4732663.

[44] WenyuWu, Xiumei Fan, Yaqiong Xue, and Yusheng Huang. An attention mechanismbased

**Research Article**

method for predicting traffic flow by gcn. In 2021 40th Chinese Control Conference (CCC), pages 8410–8415, 2021. doi: 10.23919/CCC52363.2021.9550673.

bibliography 39

[45] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1):4–24, 2021.

[46] Guo Xiaojian and Zhu Quan. A traffic flow forecasting model based on bp neural network. In 2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), volume 3, pages 311–314, 2009.

[47] Lakshmi Yermal and P. Balasubramanian. Application of auto arima model for forecasting

returns on minute wise amalgamated data in nse. In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pages 1–5, 2017. [48] Zhou Yu, Xingyu Shi, and Zhaoning Zhang. A multi-head self-attention transformerbased model for traffic situation prediction in terminal areas. IEEE Access, 11:16156– 16165, 2023.

[49] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 21(9):3848–3858, 2020.

[50] Xin Zheng, Miao Zhang, Chunyang Chen, Chaojie Li, Chuan Zhou, and Shirui Pan. Multi-relational graph neural architecture search with fine-grained message passing. In 2022 IEEE International Conference on Data Mining (ICDM), pages 783–792, 2022.

[51] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. AI open, 1:57–81, 2020.

[52] Jiawei Zhu, Chao Tao, Hanhan Deng, Ling Zhao, Pu Wang, Tao Lin, and Haifeng Li. Ast-gcn: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting, 2020.