Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

LSTM variants meet graph neural networks for road speed prediction

Zhilong Lu[a], Weifeng Lv[a,b], Yabin Cao[a], Zhipu Xie[a], Hao Peng[a,b], Bowen Du[a,b,*]

[a] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China*
[b] *Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, 100083 China*

A R T I C L E I N F O

Article history:
Received 8 November 2019
Revised 24 February 2020
Accepted 6 March 2020
Available online xxx

Communicated by Dr. K. Li

Keywords:
Neural network
LSTM
LSTM Variant
GNN
Road speed prediction

A B S T R A C T

Traffic flow prediction is a fundamental issue in smart cities and plays an important role in urban traffic planning and management. An accurate predictive model can help individuals make reliable travel plans and choose optimal routes while efficiently helping administrators maintain traffic order. Road speed prediction, which is a sub-task of traffic flow forecasting, is challenging due to the complicated spatial dependencies characterizing road networks and dynamic temporal traffic patterns. Given the power of recurrent neural networks (RNNs) in learning temporal relations and graph neural networks (GNNs) in integrating graph-structured and node-attributed features, in this paper, we design a novel graph LSTM (GLSTM) framework to capture spatial-temporal representations in road speed forecasting. More specifically, we first present a temporal directed attributed graph to model complex traffic flow. Then, to take advantage of the structure properties and graph features, we employ a message-passing mechanism for feature aggregation and updating. Finally, we further implement several variants of LSTMs with a GN block under the encoder-decoder framework to model spatial-temporal dependencies. The experiments show that our proposed model is able to fully utilize both the road latent graph structure and traffic speed to forecast the road state during future periods. The results on two real-world datasets show that our GLSTM can outperform state-of-the-art baseline methods by up to 32.8% in terms of MAE, 43.2% in terms of MAPE and 23.1% in terms of RMSE.

1. Introduction

The accurate and real-time prediction of traffic speed is a crucial and fundamental task in the construction of smart cities and is particularly useful for many applications such as traffic network planning, route guidance, and congestion avoidance [1]. Actually, reliable road speed prediction can not only benefit city management departments by providing tools for policy making but also give individual travellers sufficient information when making plans [2]. There are numerous methods for traffic prediction based on estimating road speed, volume, and travel time. These methods can be divided into two categories: conventional shallow methods and emerging deep-learning-based models. The former methods, such as support vector machine (SVM) [3] and the autoregressive integrated moving average (ARIMA) model [4], can handle time-series data but cannot capture the spatial-temporal relationships of road networks. At the same time, given the increasing research interest in time-series data prediction based on deep learning methods, many neural-network-based models have been proposed in the latter category, such as the ResNet-based ST-ResNet model [5], the convolutional neural network (CNN)-based model [6], the GCN-based model [7] and other deep-learning-based traffic flow prediction methods [8,9]. These models can outperform shallow models in capturing spatial-temporal features.

In recent years, LSTMs have gained substantial interest due to their ability to model the long-term dependencies of time-series data. Many methods, such as [10–12], examine the applicability and demonstrate the advantages of LSTMs for traffic prediction. Furthermore, a variant of LSTMs has been proposed for improving performance in many tasks, such as [13–16]. Many papers were also published after [17], which gives a detailed introduction of GNNs. Given the advantages of GNNs and graph convolutional neural networks (GCNs), graph-structured data modelling has gained increased attention, especially for traffic networks such as in [18–22]. Regardless, these methods model spatial-temporal relations of road networks by constructing adjacent matrices and using convolution operations while ignoring the latent properties of road structures.

In this paper, we aim to predict road speed more accurately, especially under extreme circumstances such as following traffic accidents. However, we face several challenges when addressing

* Corresponding author.
E-mail addresses: zhilong_lu@buaa.edu.cn (Z. Lu), lwf@buaa.edu.cn (W. Lv), cybin@buaa.edu.cn (Y. Cao), xiezhipu@buaa.edu.cn (Z. Xie), penghao@act.buaa.edu.cn (H. Peng), dubowen@buaa.edu.cn, zjj@buaa.edu.cn (B. Du).

https://doi.org/10.1016/j.neucom.2020.03.031
0925-2312/

this task: (1)**Complex road structure**. There are many relations between segments in a road network, such as different directions. Moreover, segments and sections have many properties. (2)**Spatial and temporal correlation**. The adjacent sections on a road in the same direction have similar traffic speeds most of the time. However, adjacent sections in opposite directions are generally not related in terms of traffic speed. The temporal correlations of adjacent sections exhibit the same phenomena as spatial correlations.

Motivated by the idea that performance gains can be achieved by using GNNs for modelling graph-structured data and LSTM for learning long-term dependencies of time-series data, this paper proposes combining GNNs and LSTM for predicting the traffic state of the next time period for all related road segments. More specifically, a road network is constructed as a graph to capture propagation patterns, and this model takes the graph as input and returns a graph as output. The input graph has edges, nodes, and global-level attributes, while the output graph has the same structure with updated attributes. The goal of this study is to develop a new model for accurately capturing traffic state connections between adjacent sections and predicting the road speed of future periods. The contributions of this paper can be summarized as follows:

- We fully utilize the graph structure to improve the performance of road prediction by using a GNN. We can capture adjacent section relations for different directions and utilize road network properties while maintaining the original structure, especially under extreme traffic conditions.
- We propose GLSTM, which combines GN blocks with LSTM cells, and then, we adopt an encoder-decoder model to capture the spatial-temporal properties of road network structures. Furthermore, we implement several variants of GLSTM cells to prove the effectiveness of the combination of the GN and LSTM cells.
- We conduct extensive experiments on two real-world datasets and obtain accurate predictions. The state-of-the-art methods and strong baselines are compared with our method for road speed forecasting.

The remaining parts are organized as follows. First, related works are presented and compared with standard LSTM and GNNs in Section 2. Next, we give several definitions and formulate the problem in Section 3. Then, we present the details of the proposed methods in Section 4, followed by Section 5, where the experimental results are evaluated and explained. Finally, a comprehensive conclusion and future research are discussed in Section 6.

2. Related work

2.1. Traffic speed prediction

Traffic speed prediction is a critical component in an intelligent transportation system (ITS) and is a fundamental problem in urban computing. Methods for traffic speed prediction fall into two categories: parametric and non-parametric methods [23].

Many parametric methods were used for traffic speed prediction before deep learning methods appeared. These methods are called shadow methods, in contrast to deep learning models. Levin and Tsao found that the ARIMA (0,1,1) was the most statistically significant of all forecasting intervals [24]. K-nearest neighbour (KNN) algorithms were investigated for traffic prediction and achieved slightly better results than other state-of-the-art techniques [25]. Ghosh et al. [26]. proposed structural time-series model (STM) for short-term traffic condition forecasting. The above methods can capture the temporal properties of time-series data while ignoring the spatial attributes.

Given the advances of deep-learning-based methods, many non-parametric algorithms have been applied to various applications. Indeed, neural-network-based models have been commonly used for traffic speed prediction. Wang et al. [27]. proposed a deep learning method with an errorfeedback recurrent convolutional neural network structure (eRCNN) for continuous traffic speed prediction. Lv et al. [28]. proposed look-up convolution recurrent neural network (LC-RNN) to achieve more accurate traffic speed prediction by taking advantage of a rational integration of both RNN and CNN models to learn more meaningful time-series patterns that can adapt to the traffic dynamics of surrounding areas. Kim et al. [29]. demonstrated that embedding topological information of the road network improves the process of learning traffic features and then used a graph of a vehicular road network with recurrent neural networks (RNNs) to infer the interaction between adjacent road segments as well as the temporal dynamics. A neural network with capsules that replaces max pooling by dynamic routing has been proposed for traffic speed prediction [30]. These methods can incorporate many hidden unit and complex neural network frameworks to capture the temporal and spatial attributes of data.

2.2. Long short-term memory

LSTM networks are RNNs equipped with a special gating mechanism that controls access to memory cells [31]. LSTM can be designed to model long-term dependency relations, especially when handling sequential data. Liao et al. [32] proposed a hybrid model based on the Seq2Seq framework with LSTM for traffic prediction. Wei et al. [33] proposed a spatio-temporal LSTM network preceded by map matching to predict fine-grained traffic conditions. Variants of LSTM have been proposed, whereas the above studies all include standard versions. Neil et al. [14]. introduced the phased LSTM model, which extends the LSTM unit by adding a new time gate and achieved faster convergence than regular LSTM on tasks that require the learning of long sequences. Several papers have predicted traffic while considering road structure. Cui et al. [34] modelled the traffic network as a graph and proposed a novel deep learning framework, the traffic graph convolutional long short-term memory neural network (TGC-LSTM), to learn the interactions between roadways in a traffic network and forecast the network-wide traffic state. Zhao et al. [35] proposed a novel neural-network-based traffic forecasting method called the temporal graph convolutional network (T-GCN) model, which combined the graph convolutional network (GCN) and gated recurrent unit (GRU). Wu et al. [36] proposed the graph attention LSTM network (GAT-LSTM) and used it to build an end-to-end trainable encoder-forecaster model to solve the multi-link traffic flow forecasting problem. It is not efficient to capture road structure properties without graphs, and these methods treat road links as adjacent matrices.

2.3. Graph neural networks

The concept of GNNs was first introduced in 2005 [37] and further elaborated upon in 2009 [38]. GNNs have been able to achieve satisfactory results in multiple domains where data are typically structured as a graph [39]. Zhou et al. [40] noted that graphs can be used as denotations of a large number of systems across various areas, including social networks, physical systems, knowledge graphs and many other research areas. Ma et al. [41] provided a new graph convolution framework that involves every path linking the message sender and receiver with learnable weights depending on the path length, which corresponds to the maximal entropy random walk inspired by the path integral idea in physics. Moreover, unlike standard neural networks, GNNs include a state that

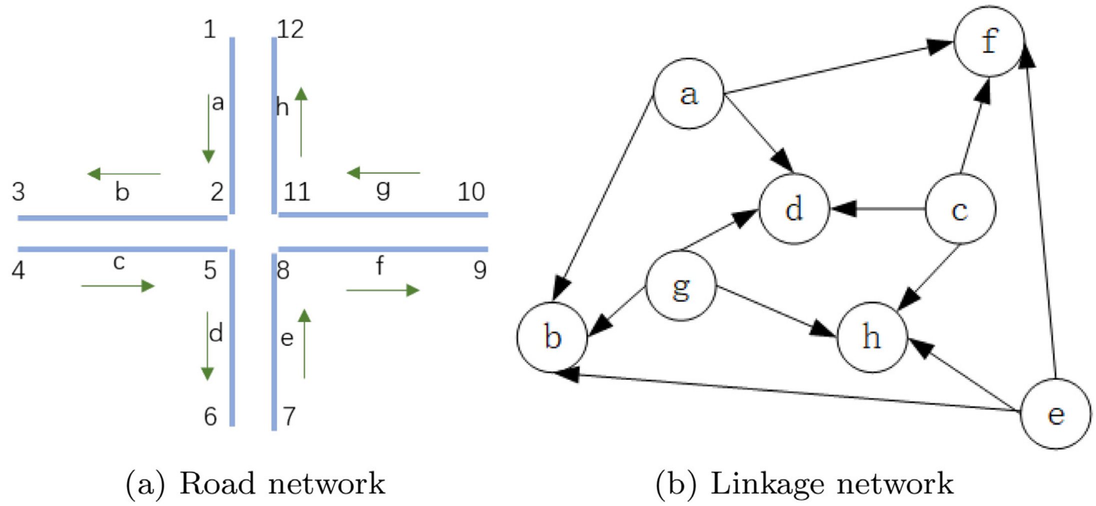


Fig. 1. Differences between two modelling structures.

can represent information from its neighbourhood with an arbitrary depth and can capture the dependence of graphs via message passing between the nodes of the graphs. Xu et al. [42] presented a theoretical framework for analysing the expressive power of GNNs to capture different graph structures and explained the power of GNNs for handling graph-structured data.

A new graph network (GN) framework was presented that defines a class of functions for relational reasoning over graph-structured representations [17]. The main unit of computation in the GN framework is the GN block, which takes a graph as input and returns a graph. Moreover, the GN block can be constructed for various complex architectures when combined with other models.

Because road networks can be naturally modelled with graph structures, many studies on traffic-prediction-based GNNs have been conducted. The graph convolution generalizes the traditional convolution to data to graph structures. Li et al. [43] modelled the spatial dependency of traffic as a diffusion process on a directed graph and proposed the diffusion convolutional recurrent neural network (DCRNN), which can capture both spatial and temporal dependencies among time series. Guo et al. [7] proposed a novel attention-based spatial-temporal graph convolutional network (ASTGCN) model to solve the traffic flow forecasting problem, therein combining the spatial-temporal attention mechanism and spatial-temporal convolution. A multi-view graph convolutional network (MVGCN) for the crowd flow forecasting problem was built using spatial graph convolution [44]. Wang et al. [45] proposed a new topological framework called a linkage network to model road networks, presented the propagation patterns of the traffic flow and designed a graph recurrent neural network (GRNN) to learn the propagation patterns in the graph.

3. Definitions and problem statement

We briefly introduce the traffic speed prediction problem in this section. First, we define several key concepts and then formulate the problem.

Graph for GNN. A normal graph is composed of nodes and edges that connect nodes. However, a graph in a GNN is defined with additional properties. Specifically, graphs are defined as $G = (V, E, V^{attr}, E^{attr}, U^{attr})$ in a GNN. $V = v_j$ represents all nodes of a graph, while $E = (e_k, s_k, r_k)$ contains all the edges. n_j and e_k are small integers sorted in ascending order. The sender and receiver of an edge are nodes in V and are represented by s_k and r_k, respectively. V^{attr} contains the attributes of all nodes. The i-th row of V^{attr} is the attribute vector of the i-th node. Similarly, E^{attr} is a matrix that contains the attributes of all edges. The i-th row of E^{attr} represents the attribute vector of the i-th edge. Finally, U^{attr} represents the attribute vector of the whole graph.

Graph isomorphism. Given graphs $G_1 = \{V_1, E_1\}$ and $G_2 = \{V_2, E_2\}$, they are isomorphic under the following conditions: 1) there is a bijection (one-to-one correspondence) f from V_1 to V_2, and 2) there is a bijection g from E_1 to E_2 that maps each edge (v, u) to $(f(v), f(u))$.

Road network. We model a road network as a directed graph $G = (V, E)$. Each vertex $v \in V$ denotes an intersection or the segmentation point of a road, and each edge $e \in E$ represents a road segment that links the contiguous intersections. A directional edge $e \in E$ from intersection v_i to v_j is established when there is a road segment between v_i and v_j. $V = \{v_1, v_2, \cdots, v_N\}$, where N is the number of road intersections. $E = \{e_1, e_2, \cdots, e_M\}$, where M is the number of road segments. Fig. 1(a) illustrates the graph structure of a road network.

Linkage graph. For handling a simple graph and combining the useful information contained in vertices and edges, the linkage graph-structured network [45] is adopted, which is different compared to a normal road network. $G^* = (V^*, E^*)$ denotes the new linkage graph. Each vertex $v^* \in V^*$ denotes a segment of a road, and each edge $e^* \in E^*$ represents a direct link between adjacent segments. $V^* = \{v_1^*, v_2^*, \cdots, v_M^*\}$, where M is the number of road segments. $E^* = \{e_1^*, e_2^*, \cdots, e_L^*\}$, where L is the number of linkages. A directional edge e_i^* from segment v_i^* to v_j^* will be established if there is an approachable path from v_i^* to v_j^*. Compared with G defined above, each node in G^* denotes a road segment $e \in E$, and each edge denotes a reachable direct link between $e \in E$. Fig. 1(b) illustrates the graph structure of the linkage network in Fig. 1(a). In this paper, the road traffic speed graph (TSG) is a linkage graph whereby each vertex contains information about the traffic state of the road segment between two intersections, and the TSG will be fed to the proposed model. Additional information is considered when constructing a linkage graph. More specifically, the length and width of the segment for each node, as well as the traffic light and turning direction of each edge are included.

Traffic speed prediction. Given the data of road segments in the past time intervals from t_{j-2k} to t_{j-1}, the problem is to predict the road traffic speed of road segments v_i^* during peak hours from t_j to t_{j+k} for several connected road segments simultaneously. We aim to predict $Y = \{S_{t_r}^{v_i^*} \mid r = j, \cdots, j+k\}$ given the historical data

Please cite this article as: Z. Lu, W. Lv and Y. Cao et al., LSTM variants meet graph neural networks for road speed prediction, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.03.031

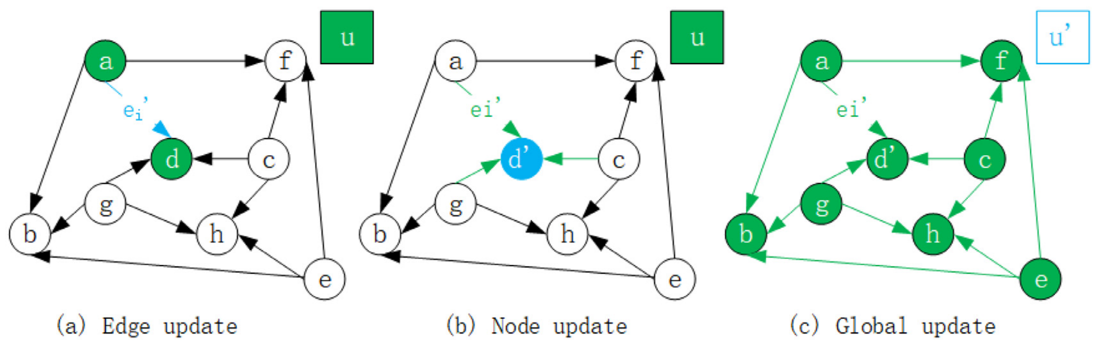


Fig. 2. Updates in a GN block. (*a*) denotes how the edge updates in a GN block: it updates only the edge and related node properties. (*b*) denotes how the node updates: it only updates the related edges. (*c*) denotes how the graph updates: it updates all the edges and nodes.

$\{S_{t_r}^{v_i^*} \mid r = j-2k, \cdots, j-1\}$, where $i \in \{1, 2, \cdots, M\}$, k represents the smallest time interval, $S_{t_j}^{v_i^*}$ denotes the traffic speed of road segment v_i^* at t_j, while $\hat{S}_{t_j}^{v_i^*}$ is to be predicted. As Fig. 1(a) shows, a to h are different road segments, and they are connected by road intersections 2, 5, 8, and 11. Our goal, expressed in Eq. (1), is used to predict the road speed of all eight segments from t_j to t_{j+k} given past road speeds from t_{j-2k} to t_{j-1}.

$$\left(\hat{S}_{t_j}^{v_i^*}, \cdots, \hat{S}_{t_{j+k}}^{v_i^*}\right) = \mathcal{F}\left(S_{t_{j-2k}}^{v_i^*}, \cdots, S_{t_{j-1}}^{v_i^*}\right) \tag{1}$$

4. Proposed method

We now describe the proposed GLSTM model, which combines LSTM and GNNs to predict road traffic speed. First, we formulate the GN block. Then, we introduce the proposed graph LSTM cell modified from the LSTM cell. Subsequently, we explain the entire architecture of the proposed model in detail. Finally, we choose the optimal loss functions and optimize the training process.

4.1. GN block for graph to graph

A GN block, as defined in Eq. (2), is a “graph-to-graph” module, which means that it takes a graph as input, denoted as $G = (V, E, V^{attr}, E^{attr}, U^{attr})$, and returns a graph denoted as $G' = (V', E', V^{attr'}, E^{attr'}, U^{attr'})$ as output.

$$G' = \mathbf{GN}(G) \tag{2}$$

Graphs G' and G are isomorphic, which means that $V' = V$ and $E' = E$.

When G is provided as input to a GN block, the computations proceed from the edge to the node and to the global level. Fig. 2 illustrates that graph elements are involved in each of the computation steps. The computation in a GN block is described in Algorithm 1. In the GN block, $V^{attr'}$ is considered to be associated with V^{attr}, as well as E^{attr} and U^{attr}. These additional attributes are merged into a new matrix, denoted as $\widetilde{E}^{attr}$. The i-th row in $\widetilde{E}^{attr}$, denoted as $\widetilde{E}_i^{attr}$, is produced by

$$\widetilde{E}_i^{attr} = [E_i^{attr}, V_r^{attr}, V_s^{attr}, U^{attr}] \tag{3}$$

where V_r^{attr} and V_s^{attr} denote the attributes of the receiver and sender nodes of the i-th edge, respectively. The right-hand side of Eq. (3) means concatenating all of these vectors into a longer vector.

$E^{attr'}$ is then produced with $\widetilde{E}^{attr}$ by a one-layer fully connected neural network.

$$E^{attr'} = tanh(W_E * \widetilde{E}^{attr} + b_E) \tag{4}$$

Algorithm 1 Computation in a GN block.

Input: $G = (V, E, V^{attr}, E^{attr}, U^{attr})$
Output: $G' = (V', E', V^{attr'}, E^{attr'}, U^{attr'})$

1: **for** i in number of E **do**
2: $\quad \widetilde{E}_i^{attr} = [E_i^{attr}, V_r^{attr}, V_s^{attr}, U^{attr}]$
3: **end for**
4: $E^{attr'} = tanh(W_E * \widetilde{E}^{attr} + b_E)$
5: **for** i in number of V **do**
6: $\quad \widetilde{V}_i^{attr} = [agg_{e\to v}^i(E^{attr'}), V_i^{attr}, U^{attr}]$, where $agg_{e\to v}^i(E^{attr'}) = \sum_{j\in\{j|E_j.r_k=V_i\}} E_j^{attr'}$
7: **end for**
8: $V^{attr'} = tanh(W_V * \widetilde{V}^{attr} + b_V)$
9: $\widetilde{U}^{attr} = [\sum E^{attr'}, \sum V^{attr'}, U^{attr}]$
10: $U^{attr'} = tanh(W_U * \widetilde{U}^{attr} + b_U)$

where $tanh(*)$ is the hyperbolic tangent function, W_E are weights of $\widetilde{E}^{attr}$ and b_E are biases.

The i-th row of $\widetilde{V}^{attr}$ is defined as

$$\widetilde{V}_i^{attr} = [agg_{e\to v}^i(E^{attr'}), V_i^{attr}, U^{attr}] \tag{5}$$

where

$$agg_{e\to v}^i(E^{attr'}) = \sum_{j\in\{j|E_j.r_k=V_i\}} E_j^{attr'} \tag{6}$$

and Σ means aggregating many vectors into one vector by summing them up with vector addition. For simplicity, Eq. (6) indicates aggregating attributes of edges whose receivers are the i-th node, denoted as V_i.

Similarly, $V^{attr'}$ is also produced by a one-layer fully connected neural network with $\widetilde{V}^{attr}$, where b_V and b_U are biases for $V^{attr'}$ and $U^{attr'}$ correspond.

$$V^{attr'} = tanh(W_V * \widetilde{V}^{attr} + b_V) \tag{7}$$

Finally, $U^{attr'}$ is produced as follows:

$$U^{attr'} = tanh(W_U * \widetilde{U}^{attr} + b_U) \tag{8}$$

where

$$\widetilde{U}^{attr} = \left[\sum E^{attr'}, \sum V^{attr'}, U^{attr}\right] \tag{9}$$

4.2. Graph LSTM cell

Inspired by regular LSTM cells, in this paper, GLSTM cells, as shown in Fig. 3, are constructed to handle graph sequences rather than sequential vectors. Similar to regular LSTM cells, GLSTM cells maintain two hidden states, GC and Gh, which are both represented as graphs. In each step of a GLSTM model, the GLSTM cell

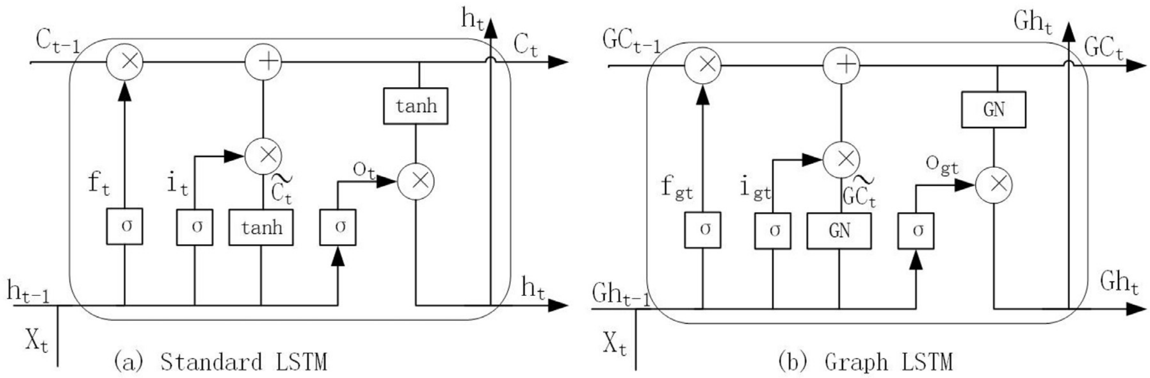


Fig. 3. Inner structure of graph LSTM cell.

takes GC_{t-1} and Gh_{t-1} as input, where GX_t is also included as input, and returns GC_t and Gh_t as output. A GLSTM cell can be formulated as follows:

$$(GC_t, Gh_t) = GLSTM(GC_{t-1}, Gh_{t-1}, GX_t) \tag{10}$$

The forget gate mechanism of GLSTM is indicated as f_t in Fig. 3 and is formulated as follows:

$$f_t = \sigma(GN_f([Gh_{t-1}, GX_t])) \tag{11}$$

In this part, the operations are slightly different from the regular operations. For example, in Eq. (11), $G_M = [G_1, G_2]$ means the concatenating of two graphs, which is defined as follows:

$$\begin{aligned} G_M.V &= G_1.V = G_2.V \\ G_M.E &= G_1.E = G_2.E \\ G_M.V^{attr} &= [G_1.V^{attr}, G_2.V^{attr}] \\ G_M.E^{attr} &= [G_1.E^{attr}, G_2.E^{attr}] \\ G_M.U^{attr} &= [G_1.U^{attr}, G_2.U^{attr}] \end{aligned} \tag{12}$$

Similarly, an activation function, such as the *sigmoid* function for graphs, denoted as $\hat{G} = \sigma(G)$, is defined as follows:

$$\begin{aligned} \hat{G}.V &= G.V \\ \hat{G}.E &= G.E \\ \hat{G}.V^{attr} &= \sigma(G.V^{attr}) \\ \hat{G}.E^{attr} &= \sigma(G.E^{attr}) \\ \hat{G}.U^{attr} &= \sigma(G.U^{attr}) \end{aligned} \tag{13}$$

GN_f is a GN block for generating the forget gate.

The input gate, denoted as i_t, is defined as

$$i_t = \sigma(GN_i([Gh_{t-1}, GX_t])) \tag{14}$$

where GN_i is a GN block for generating the input gate.

Based on the forget and input gates, the long-term memory is updated by forgetting the meaningless old memories and adding new memories:

$$GC_t = f_t \odot GC_{t-1} + i_t \odot NM_t \tag{15}$$

where

$$NM_t = GN_{NM}([Gh_{t-1}, GX_t]) \tag{16}$$

and the $\odot$ operation means the dot product for graphs.

The output gate is defined as follows:

$$o_t = \sigma(GN_o([Gh_{t-1}, GX_t])) \tag{17}$$

The Gh_t is updated based on o_t and defined as follows:

$$Gh_t = o_t \odot GN_o([Gh_{t-1}, GX_t]) \tag{18}$$

4.3. Model framework

Based on the GLSTM cell, we propose a new model that combines the LSTM and GN frameworks for handling graph-structured data to predict the traffic speed of a road network. As Fig. 4 shows, we can see that GLSTM has two components: the encoder component and the decoder component. The encoder component feeds a variable-length source graph sequence to GLSTM while also mapping the graph-structured data to fixed-length data. Then, the decoder component outputs graph-structured data from GLSTM while also mapping fixed-length data to a variable-length target graph sequence. The EOS of Fig. 4 just represent the beginning and ending of decoder component. Finally, the entire architecture can be applied to sequential graph-structured data while maintaining the original structure and capturing latent properties for accurate road speed prediction.

4.4. Objective function and optimization

The objective function is used to define the loss between predicted values and true values. In this model, we choose the mean squared error (MSE) as the loss function,

$$MSE = \frac{1}{T}\sum_{i=1}^{T}(v_t - \tilde{v}_t)^2 \tag{19}$$

where $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \frac{1}{M}\frac{1}{T}\sum_{i=1}^{M}\sum_{j=1}^{T} \| \hat{S}_{t_j}^{v_i^*} - S_{t_j}^{v_i^*} \|^2 \tag{20}$$

where T is the time span to be predicted, and M is the road segments for training and testing. Furthermore, $\hat{S}_{t_j}^{v_i^*}$ is the predicted road speed of a specific road segment v_i^* at t_j, while $S_{t_j}^{v_i^*}$ is the real road segment. We also select the Adam optimizer as the optimization algorithm in the training process.

5. Experiments

5.1. Dataset description

In this paper, we test our proposed models on two real-world datasets. One dataset is collected from Beijing city, which is the capital of China, while the other dataset is collected from Xi'an city, a southern city of China.

5.1.1. Beijing traffic network

The data contain two parts: the road network structure of Beijing and the taxi speed. The selected time periods of the trajectory

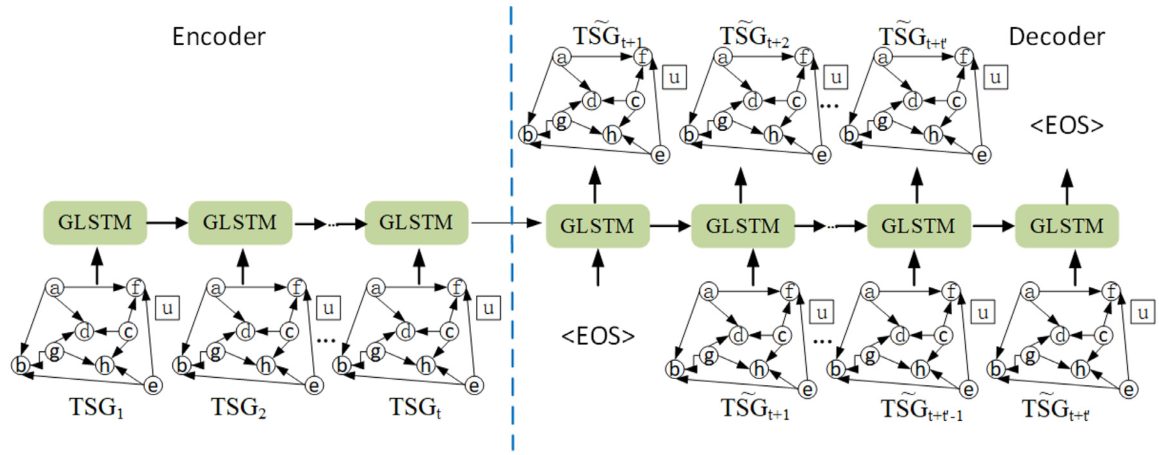


Fig. 4. Framework of the GLSTM model.

data range from Aug. 01 2018 to Oct. 31 2018, representing 85 days in total since several days were missing. For simplicity, we selected 69 road segments for the experiments, as shown in Fig. 5(a). We chose 80% of the data as the training set, while the remainder was used as the testing set. In this way, 68*69=4692 samples for the training set and 17*69=1173 samples for testing were utilized.

5.1.2. Xi'an traffic network

The data contain the road network structure of Xi'an and taxi speed data. We choose the time periods of trajectory data ranging from Sept. 01 2017 to Nov. 30 2017, 91 days totally. For simplicity, we select 28 road segments in this experiment, as shown in Fig. 5(b) and Fig. 5(c). Data from Nov. 17 2017 to Nov. 30 2017 is used for testing, while the others are used for training. In this way, we chose $77 \times 28 = 2156$ samples as the training set and $14 \times 28 = 392$ samples for testing.

5.2. Preprocessing

We need to calculate the road speed of selected segments from taxi speeds. First, we calculate the traffic speed from the original taxi trajectory data, and ST-Matching [46] is applied to match the low-sampling-rate GPS trajectories with the road network. The traffic speed value of this road segment is missing when no taxi passed through a certain road segment during a time interval. Then, we use the moving average method to address the missing values of road segments when faced with incomplete data.

We also need to construct a linkage graph based on map data for training. First, we select the specific road and find all related segments. And the selected segments will be plotted clearly by using Echarts[1]. Then we find the connections of these segments and add additional information of segments based on property files. Finally, we tag every segments an new id and save all connections between them with personal properties.

In our experiments, we aggregated the traffic speed into 5-minute windows and applied Z-score standardization for data preprocessing. We only trained and predicted the traffic speed of a specific period during the morning peak hours from 6:00 to 9:00 because there were sufficient vehicles on the road and because the data are of high quality. We choose two hours, from 6:00 to 8:00, as the observation period for training and predict the next one-hour period, from 8:00 to 9:00; thus, the length of an input sequence is 24, while the output length is 12. In addition, we incorporate some basic properties of road segments and intersections to capture road network structures such as the width, length and type of road segment.

5.3. Evaluation metrics

The mean absolute error (**MAE**), mean absolute percentage error (**MAPE**) and root mean square error (**RMSE**) are used to evaluate the performance of the proposed method and the comparative approaches. These metrics are defined as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|v_t - \tilde{v}_t| \tag{21}$$

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{v_t - \tilde{v}_t}{v_t}\right| \tag{22}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(v_t - \tilde{v}_t)^2} \tag{23}$$

where v_t and $\tilde{v}_t$ are the real and predicted traffic speeds at time t, respectively.

5.4. Baselines

In this section, we compare the proposed model with several baselines. Furthermore, we compare modified variants of LSTM combined with the GN block and prove that the GN block can improve the performance of the LSTM method when combined with the LSTM method.

- **Multilayer perceptron (MLP)**: In our experiments, the MLP utilized three fully connected layers consisting of 24, 24 and 12 hidden units.
- **Seq2Seq** [47]: This model uses an RNN to encode the input sequences into a feature representation and another RNN to generate predictions.
- **GRU** [13]: GRUs are a gating mechanism in RNNs. The GRU is similar to an LSTM with a forget gate but has fewer parameters than LSTM because it lacks an output gate. GRUs have been shown to exhibit even better performance on certain datasets.
- **Phased LSTM** [14]: Phased LSTM extends the LSTM unit by adding a new time gate. This gate is controlled by a parameterized oscillation with a frequency range that produces updates of the memory cell only during a small portion of the cycle.

[1] https://www.echartsjs.com/zh/index.html

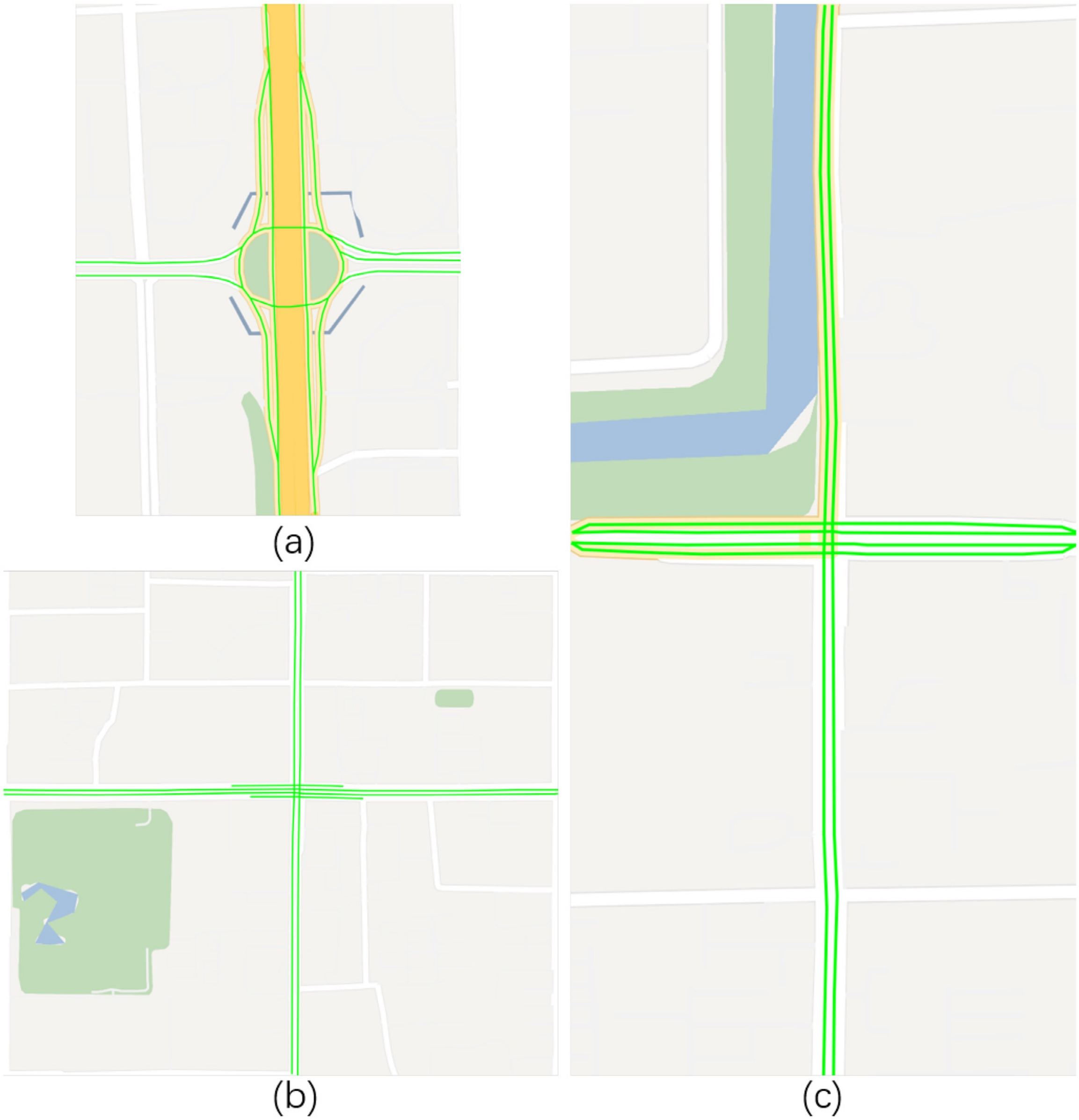


Fig. 5. Selected road networks. (*a*) denotes a road intersection in Beijing that has 69 segments. (*b*) and (*c*) represent two different intersections in Xi'an. (*b*) is composed of 14 segments, while (*c*) has the same segments.

- **NLSTM** [15]: NLSTM is a simple extension of the LSTM model that adds depth via nesting, as opposed to via stacking. Nested LSTMs outperform both stacked and single-layer LSTMs with similar numbers of parameters on various character-level language modelling tasks, and the inner memories of an LSTM learn longer term dependencies compared with the higher level units of a stacked LSTM.
- **IndyLSTM** [16]: IndyLSTM differs from regular LSTM cells in that the recurrent weights are not modelled as a full matrix but rather as a diagonal matrix, i.e., the output and state of each LSTM cell depend on the inputs and their own outputs, as opposed to the inputs and outputs of all the cells in the layer.
- **GNN** [17]: A simple GNN model that uses GN blocks for graph-structured data to predict traffic speed.
- **T-GCN** [35]: The temporal graph convolutional network (T-GCN) model is a novel neural-network-based traffic forecasting method and uses a combination of a GCN and GRU to capture the spatial and temporal dependence simultaneously.
- **DCRNN** [43]: DCRNN is a deep learning framework for traffic forecasting that incorporates both spatial and temporal dependencies in the traffic flow. DCRNN captures the spatial dependency using bidirectional random walks on the graph and the temporal dependency using the encoder-decoder architecture with scheduled sampling.

5.5. Experimental settings

All experiments were complied and tested on a cluster with four NVIDIA TITAN Xp CPUs. For simplicity, we select several road segments for the experiments as discussed before in the data description. In addition, early stopping behaviour is exploited to accelerate inference and terminate the training process when the error on the validation set increases. These neural-network-based approaches were implemented using TensorFlow 1.9.0 and trained using the Adam optimizer with learning rate annealing. We execute grid search strategy to find the optimal hyper-parameters on

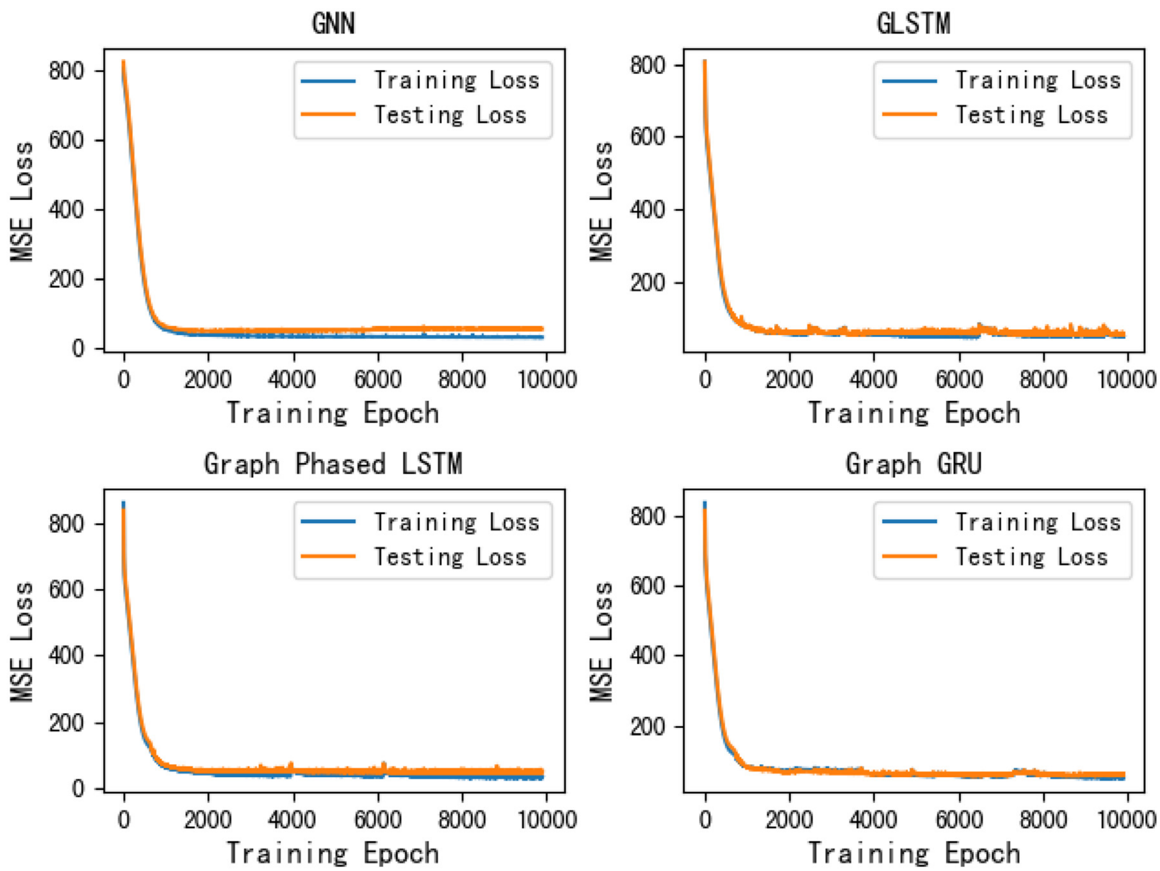


Fig. 6. Training and testing loss. Shows the training and testing losses of four GNN models with different training iteration numbers.

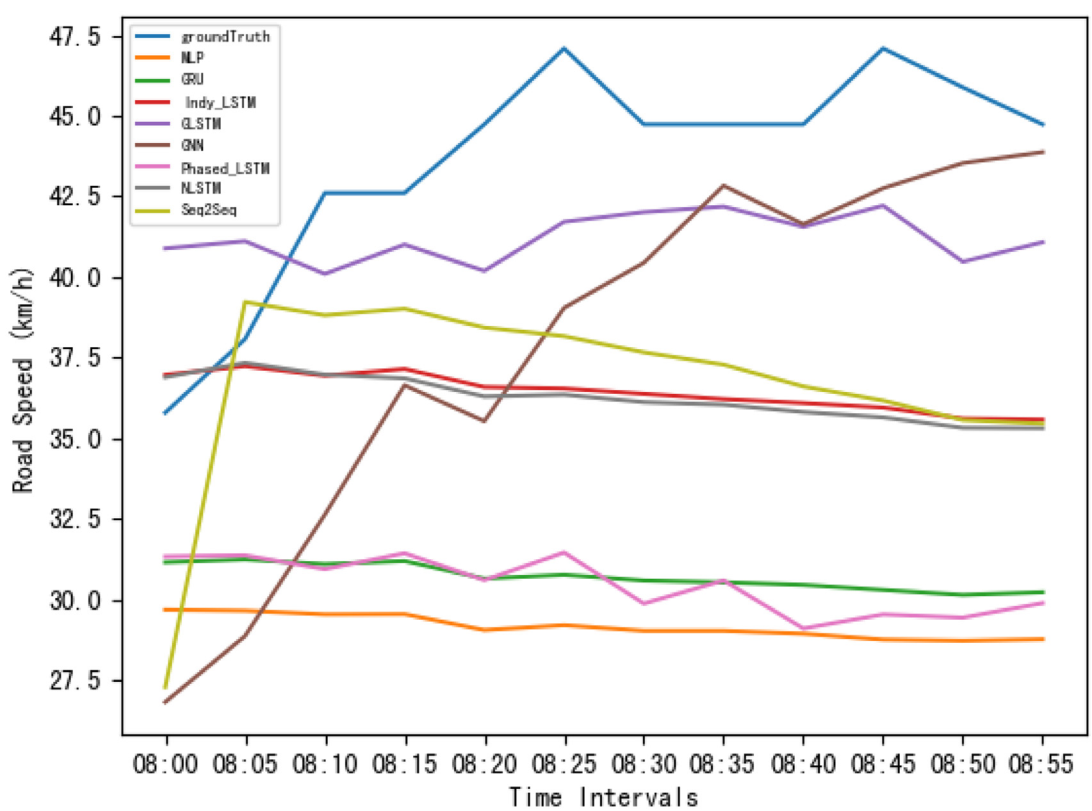


Fig. 7. Predicted results of Xi'an. Denotes the predicted traffic speed between the ground truth and selected baselines.

the validation dataset. Additionally, we set *MSE* as the loss function when training models. The historical time window is 2 hours, and the predicted time window is 1 h. Almost these deep learning methods are based on the encoder-decoder framework, we set the input length as 24 ($T = 24$), while the length of output is set to 12 ($T = 12$).

5.6. Results and discussion

We present the comparison results with the baselines, where all of these LSTMs are variations of LSTMs, therein incorporating changes from standard LSTM cells in Table 1. There are two real-world datasets for Xi'an and Beijing with three evaluation metrics. We find that the same methods trained on different datasets obtain different performances. Specifically, these models trained on the Xi'an data can achieve better results than on Beijing due to the data quality. From the table, we can easily observe that the GNN can outperform simple deep learning methods such as LSTM and MLP. These simple methods did not incorporate the spatial topology of the road network. In contrast, the GNN model can capture the graph structure and utilize latent properties to model temporal and spatial dependencies to improve performance. Furthermore, we compare state-of-the-art methods, such as T-GCN and DCRNN and both models constructed a graph as an adjacent matrix to characterize inner graph properties. The results in Table 1 show that these methods cannot model the graph structure and achieve worse performance than GNN, which models road networks as a directed graph.

Table 1
Comparison with different baselines.

Model	Xi'an			Beijing		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
MLP	7.331	0.519	8.857	11.981	0.931	14.791
Seq2Seq [47]	7.054	0.502	8.788	11.772	0.902	14.599
GRU [13]	5.972	0.431	7.753	10.731	0.841	13.552
Phased LSTM [14]	5.783	0.421	7.593	10.645	0.827	13.497
NLSTM [15]	5.716	0.412	7.491	10.628	0.816	13.275
IndyLSTM [16]	5.707	0.387	7.307	10.547	0.789	13.063
DCRNN [43]	5.457	0.369	7.173	10.346	0.775	12.841
T-GCN [35]	5.286	0.355	7.018	10.283	0.768	12.812
GNN [17]	5.182	0.348	6.903	10.159	0.742	12.725
GLSTM	**5.026**	**0.336**	**6.882**	**10.084**	**0.721**	**12.698**
Graph GRU	**4.937**	**0.302**	**6.831**	**10.027**	**0.707**	**12.631**
Graph Phased LSTM	**4.921**	**0.295**	**6.814**	**9.956**	**0.665**	**12.558**

From Table 1, we can see that several LSTM variants, such as GRU, Phased LSTM, IndyLSTM and NLSTM, can make improvements over standard LSTM. All of these variants mainly changed the inner structure of the LSTM cells to better capture long- and short-term memories. Additionally, we realize these variants of LSTMs combined with the GN block to predict traffic speed. The GN block fully

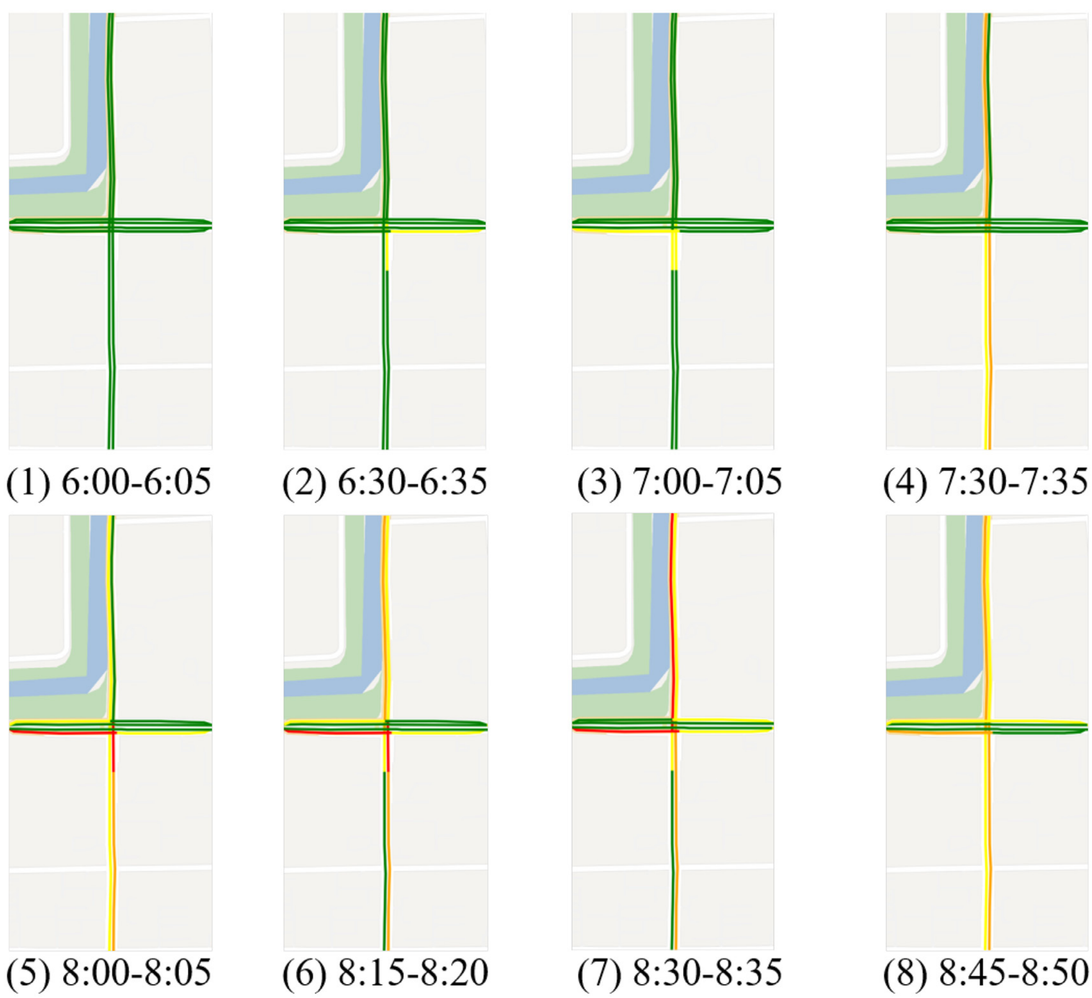


Fig. 8. Traffic speed of the road network in Xi'an during different periods. The first row denotes the traffic speed from 6:00 to 8:00. We select four time slices for the demonstration. Based on these data, we predict the next hour, from 8:00 to 9:00, and the results of the four selected periods are elaborated on the second row.

Please cite this article as: Z. Lu, W. Lv and Y. Cao et al., LSTM variants meet graph neural networks for road speed prediction, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.03.031

Fig. 9. Traffic speed of the road network in Xi'an during different periods. The first row denotes the traffic speed from 6:00 to 8:00. We select four time slices for the demonstration. Based on these data, we predict the next hour, from 8:00 to 9:00, and the results of the four selected periods are elaborated on the second row.

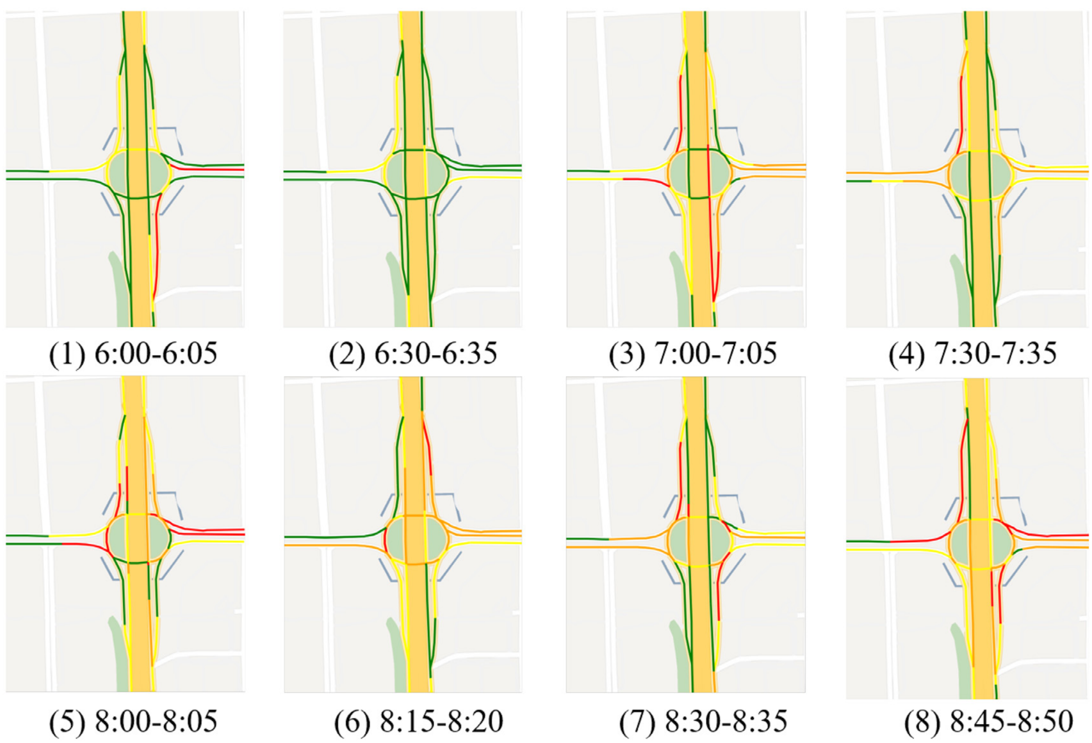


Fig. 10. Traffic speed of the road network in Beijing during different periods. The first row denotes the traffic speed from 6:00 to 8:00. We select four time slices for the demonstration. Based on these data, we predict the next hour, from 8:00 to 9:00, and the results of the four selected periods are elaborated on the second row.

utilizes the attributes of nodes and edges via aggregation and update mechanism. The results indicate that these new graph LSTM structures can outperform the unchanged LSTM models such as Graph GRU and Graph Phased LSTM. This is an effective evidence that the GNN with LSTM cells is capable of capturing road network properties and addressing traffic prediction problems; however, a huge improvement is not achieved. We also show the training process under training and testing losses of four models from high levels to low levels until they converge in Fig. 6. The loss is at a high level, which means that the performance is bad when the training begins. After a sufficient number of iterations, the MSE loss reaches a relatively low level, which represents an acceptable result.

To better understand the comparative results between the proposed methods and selected baselines, we choose a random road segment during the morning peak ranging from 8:00 to 9:00 to highlight certain aspects of the results in Fig. 7. We find that the GLSTM can capture the changing trend of road speeds on a specific road segment and track the ground truth more closely. Simultaneously, these baselines cannot accurately predict road speed because the road structures are ignored. Moreover, we illustrate the evolution of the road speed of Xi'an and Beijing road during the morning peak from 6:00 to 9:00 in Fig. 8, 9 and 10. The time interval from 6:00 to 8:00 is the real traffic speed calculated from historical taxi trajectory data, as shown in (1), (2), (3) and (4). The predicted results are shown in (5), (6), (7) and (8). We can clearly note that the related road segments have similar road speeds. The GLSTM can capture the road structures, as well as the complexity relations between these segments.

6. Conclusions and future work

A novel traffic speed prediction model called GLSTM is proposed in this paper. Inspired by the GN framework, we first construct a topological graph of the road network and feed the whole graph into the neural network and obtain a graph as output. Moreover, we combine the LSTM and GN block to build a new model to predict the traffic speed, where an encoder-decoder architecture is used. Specifically, each road segment is formulated as a vertex, and each edge denotes a connection between two segments. The properties of each vertex and edge are also considered during the training process. Then, we conduct extensive experiments on two real-world datasets obtained from Xi'an and Beijing. The results show that GLSTM can achieve the best performance against several baselines in terms of MAE, MAPE, RMSE and MSE. Moreover, variations of LSTMs with GN block can achieve improved prediction accuracy compared with standard LSTM. We can conclude that the GN block has the ability to capture graph structure properties and improve the performance of standard LSTM due to the aggregation and update mechanisms of nodes and edges attributes.

In future work, we will investigate other external factors, such as the weather and holidays, to improve the prediction accuracy. In addition, we will incorporate additional road network properties for training a standard model. The proposed model will also be applied to other spatial-temporal forecasting tasks such as traffic event prediction and traffic state estimation of longer periods. We can also explore modifying additional neural networks and feed graphs as input to improve the performance of the model. The time and memory requirements are two main problems that should be addressed in future work. It would also be meaningful to explore the explainability of GNNs.

Declaration of interests

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

CRediT authorship contribution statement

Zhilong Lu: Writing - original draft, Methodology. **Weifeng Lv:** Writing - review & editing. **Yabin Cao:** Writing - review & editing, Visualization. **Zhipu Xie:** Investigation, Validation. **Hao Peng:** Methodology. **Bowen Du:** Writing - review & editing.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments on this research work. This work is supported by the National Natural Science Foundation of China under Grant No. 51822802, 51778033, and U1811463, the Science and Technology Major Project of Beijing under Grant No. Z171100005117001.

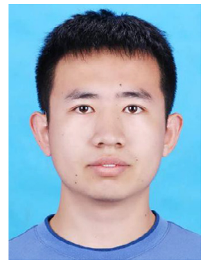
References

[1] B. Liao, J. Zhang, M. Cai, S. Tang, Y. Gao, C. Wu, S. Yang, W. Zhu, Y. Guo, F. Wu, Dest-resnet: A deep spatiotemporal residual network for hotspot traffic speed prediction, in: 2018 ACM Multimedia Conference on Multimedia Conference, ACM, 2018, pp. 1883–1891.
[2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: asurvey, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1624–1639.
[3] C.-H. Wu, J.-M. Ho, D.-T. Lee, Travel-time prediction with support vector regression, IEEE Trans. Intell. Transp. Syst. 5 (4) (2004) 276–281.
[4] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.
[5] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[6] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.
[7] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 33, 2019, pp. 922–929.
[8] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 865–873.
[9] S. Wang, J. Cao, P.S. Yu, Deep learning for spatio-temporal data mining: a survey, arXiv:/1906.04928(2019).
[10] R. Fu, Z. Zhang, L. Li, Using LSTM and GRU neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 2016, pp. 324–328.
[11] Z. Zhao, W. Chen, X. Wu, P.C. Chen, J. Liu, LSTM network: a deep learning approach for short-term traffic forecast, IET Intel. Transport Syst. 11 (2) (2017) 68–75.
[12] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, Y. Liu, Deep learning: a generic approach for extreme condition traffic forecasting, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SIAM, 2017, pp. 777–785.
[13] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv:/1412.3555(2014).
[14] D. Neil, M. Pfeiffer, S.-C. Liu, Phased LSTM: accelerating recurrent network training for long or event-based sequences, in: Advances in Neural Information Processing Systems, 2016, pp. 3882–3890.
[15] J.R.A. Moniz, D. Krueger, Nested LSTMS, arXiv:/1801.10308(2018).
[16] P. Gonnet, T. Deselaers, INDYLSTMS: independently recurrent LSTMS, arXiv:/1903.08023(2019).
[17] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, arXiv:/1806.01261 (2018).
[18] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, D.-Y. Yeung, Gaan: gated attention networks for learning on large and spatiotemporal graphs, arXiv:/1803.07294 (2018).
[19] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, arXiv:/1709.04875 (2017).
[20] Z. Cui, K. Henrickson, R. Ke, Y. Wang, High-order graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting, arXiv:/1802.07007 (2018).
[21] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, arXiv:/1906.00121 (2019).
[22] L. Bai, L. Yao, S. Kanhere, X. Wang, Q. Sheng, et al., Stg2seq: spatial-temporal graph to sequence model for multi-step passenger demand forecasting, arXiv:/1905.10069 (2019).
[23] L. Lin, J. Li, F. Chen, J. Ye, J. Huai, Road traffic speed prediction: a probabilistic model fusing multi-source data, IEEE Trans. Knowl. Data Eng. 30 (7) (2018) 1310–1323.
[24] M. Levin, Y.-D. Tsao, On forecasting freeway occupancies and volumes (abridgment), Transp. Res. Rec. (773) (1980).
[25] Y. Chen, Y. Zhang, J. Hu, Multi-dimensional traffic flow time series analysis with self-organizing maps, Tsinghua Sci. Technol. 13 (2) (2008) 220–228.
[26] B. Ghosh, B. Basu, M. O'Mahony, Multivariate short-term traffic flow forecasting using time-series analysis, IEEE Trans. Intell. Transp. Syst. 10 (2) (2009) 246.
[27] J. Wang, Q. Gu, J. Wu, G. Liu, Z. Xiong, Traffic speed prediction and congestion source exploration: a deep learning method, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 499–508.
[28] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, X. Zhou, LC-RNN: a deep learning model for traffic speed prediction., in: IJCAI, 2018, pp. 3470–3476.
[29] Y. Kim, P. Wang, L. Mihaylova, Structural recurrent neural network for traffic speed prediction, arXiv:/1902.06506 (2019).
[30] Y. Kim, P. Wang, Y. Zhu, L. Mihaylova, A capsule network for traffic speed prediction in complex road networks, in: 2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF), IEEE, 2018, pp. 1–6.
[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[32] B. Liao, J. Zhang, C. Wu, D. McIlwraith, T. Chen, S. Yang, Y. Guo, F. Wu, Deep sequence learning with auxiliary information for traffic prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 537–546.
[33] X. Wei, J. Li, Q. Yuan, K. Chen, A. Zhou, F. Yang, Predicting fine-grained traffic conditions via spatio-temporal LSTM, Wireless Commun. Mobile Comput. 2019 (2019).
[34] Z. Cui, K. Henrickson, R. Ke, Y. Wang, Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting (2018).

[35] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-GCN: a temporal graph convolutional network for traffic prediction, IEEE Trans. Intell. Transp. Syst. (2019), doi:10.1109/TITS.2019.2935152.
[36] T. Wu, F. Chen, Y. Wan, Graph attention LSTM network: a new model for traffic flow forecasting, in: 2018 5th International Conference on Information Science and Control Engineering (ICISCE), IEEE, 2018, pp. 241–245.
[37] M. Gori, G. Monfardini, F. Scarselli, A new model for learning in graph domains, in: Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, 2, IEEE, 2005, pp. 729–734.
[38] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80.
[39] P. Almasan, J. Suárez-Varela, A. Badia-Sampera, K. Rusek, P. Barlet-Ros, A. Cabellos-Aparicio, Deep reinforcement learning meets graph neural networks: an optical network routing use case, arXiv:/1910.07421 (2019).
[40] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph neural networks: a review of methods and applications, arXiv:/1812.08434 (2018).
[41] Z. Ma, M. Li, Y. Wang, Pan: path integral based convolution for deep graph neural networks, arXiv:/1904.10996 (2019).
[42] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, arXiv:/1810.00826 (2018).
[43] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: data-driven traffic forecasting, arXiv:/1707.01926 (2017).
[44] J. Sun, J. Zhang, Q. Li, X. Yi, Y. Zheng, Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks, arXiv preprint arXiv:1903.07789 (2019).
[45] X. Wang, C. Chen, Y. Min, J. He, B. Yang, Y. Zhang, Efficient metropolitan traffic prediction based on graph recurrent neural network, arXiv:/1811.00740 (2018).
[46] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, Y. Huang, Map-matching for low-sampling-rate gps trajectories, in: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2009, pp. 352–361.
[47] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

Zhilong Lu received the B.S. in Software Engineering from Northwest University, Xi'an, China, in 2014. Ph.D. degree candidate in Computer Science and Engineering from Beihang University, Beijing, China. His research interests include machine learning in intelligent transportation and city big data mining.

Weifeng Lv received the B.S. degree in Computer Science and Engineering from Shandong University, Jinan, China, and the Ph.D. degree in Computer Science and Engineering from Beihang University, Beijing, China, in 1992 and 1998 respectively. Currently, he is a Professor with the State Key Laboratory of Software Development Environment, Beihang University, Beijing, China. His research interests include smart city technology and mass data processing.



Yabin Cao received B.S. degree in Computer Science and Technology from Beihang University, Beijing, China, in 2018. Currently completing M.S degree in Beihang University, his research interests include deep temporal-spatial sequence mining and deep learning.

Zhipu Xie received the B.S. in Information and Computing Science from Central South University, Changsha, China, in 2013. Ph.D. degree candidate in Computer Science and Engineering from Beihang University, Beijing, China. His research interests include smart city technology, intelligent transportation and city big data mining.

Hao Peng is currently an assistant professor with Beijing Advanced Innovation Center for Big Data and Brain Computing in Beihang University, and School of Cyber Science and Technology in Beihang University. His research interests include representation learning, text mining and urban computing.

Bowen Du received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently an Assistant Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multi-source data fusion, and traffic data mining.