Proceedings of the 26^{th} International Conference on
Automation & Computing, University of Portsmouth,
Portsmouth, UK, 2-4 September 2021

The Sequence-to-Sequence Architecture with An Embedded Module for Long-Term Traffic Speed Forecasting with Missing Data

Ge Zheng[1], Wei Koong Chai[1], Vasilis Katos[1]
[1]Department of Computing and Informatics, Bournemouth University, Poole, Dorset, BH12 5BB, U.K.

***Abstract*—Traffic forecasting plays a crucial role in Intelligent Transportation Systems (ITSs), which is proposed to provide traffic status in advance for road users to avoid traffic congestion or other traffic incidents and for authorities to optimise the strategies of traffic management. In this paper, we develop a novel deep learning framework, based on the Sequence-to-Sequence architecture with an embedded module, for long-term traffic speed forecasting with missing data and providing high forecasting accuracy. The embedded module uses Graph Convolution Neural Network for the local spatial dependency analysis by conducting convolutional operation on the $k-hop$ neighbourhood matrix, while utilises Transformer for the global spatial dependency analysis by implementing the attention mechanism that assigns individual weights to neighbour detectors for contributing to the targeted detector. The sequence-to-sequence architecture is built to analyse temporal dependencies of the spatially-fused time series from the embedded module. To evaluate the proposed model against existing well-known ones, the real traffic speed dataset with missing data and frequent traffic incidents is used to train and test the models. The experimental results indicate that our proposed framework achieves the most accuracy forecasting, even obtaining more than 80% accuracy for forecasting two hours in advance.**

***Index Terms*—long-term, intelligent transportation system, deep learning, large-scale road networks**

I. Introduction

WITH the sharp growth of population and vehicles, transportation infrastructures are facing huge challenges, such as serious traffic congestion, increasing number of incidents and severe delay in actual travel time. These challenges bring many problems to cities like air pollution and waste of energy. To overcome the challenges and resolve these problems, the concept of Intelligent Transportation Systems (ITSs) [1] has been proposed. It aims to offer innovative services relating to different modes of transport and traffic management based on the most advanced technologies and enable road users to be better informed and safer during travelling [2]. As an important element of ITSs, traffic forecasting is to forecast traffic status for the future based on historical traffic data and then provide it for road users to improve traffic efficiency.

Many works in traffic forecasting have been reported in the literature, including studies on traffic flow, speed and travel time, since the earliest work, that used the well-known Auto-Regressive Moving Average (ARMA) model to forecast traffic volume and occupancy, was published in late 1970s [3]. For example, some variants of ARMA like Auto-Regressive Integrated Moving Average (ARIMA) [4] and Seasonal ARIMA (SARIMA) [5] were proposed for improving the capability of traffic forecasting. ARMA and its variants are statistic models and usually used for time series forecasting with linear relationship. However, [6] pointed that a non-linear relationship exists in traffic data, and ARMA and its variants are unable to completely analyse the non-linear relationship of traffic data. Owning to the wide popularity of machine learning technologies, machine learning models with non-linear kernels or activation functions have been used for analysing non-linear relationships of traffic data and achieved better performance compared to ARMA and its variants. For example, Support Vector Regression with RBF kernel (SVR) was used for traffic flow forecasting [7] while an enhanced K-Nearest Neighbor (K-NN) algorithm was also used for traffic flow forecasting [8].

With the availability of new sensor technologies and advanced big data analysis technologies, mass data with more features is available for traffic forecasting. Therefore, deep learning models that are capable of analysing big data with high dimensions have been used for traffic forecasting, instead of machine learning models that are too shallow for dealing with high dimensional data. In 2014, Huang *et al.* [9] designed a deep learning model for traffic forecasting which uses a network with Deep Belief Network (DBN) model at the bottom for unsupervised pre-training and a Multi-Task Layer (MTL) at the top for unsupervised forecasting. In the same year, Lv *et al.* [10] built a Stacked AutoEncoder model (SAE) to learn traffic flow features and trained it in a greedy layerwise fashion. Zhang [11] developed a Convolutional Neural Network (CNN) for learning traffic data as images and then used a fully-connected layer for final forecasting. The advantage of this model is that spatial dependencies can be extracted through the kernels of CNN with non-linear activation functions and its kernel size decides the area coverage of the traffic data image. However, these models are unable to analyse the temporal dependencies of traffic data. Therefore, the Recurrent Neural Network (RNN) and its variants (Long-Short Term Memory, LSTM, and Gated Recurrent Unit, GRU) [12] , that take the advantage of analysing long-term dependencies, have been combined with CNN for traffic forecasting. Wu *et al.* [13] used a 1D CNN and two LSTMs to build a hybrid deep learning model named CLTFP while Liu *et al.* [14] combined CNN with LSTM to generate a Conv-LSTM model for traffic flow forecasting.

The aforementioned models consider road networks as regular grids and traffic data having regular Euclidean structure. However, road networks are inherently irregular and traffic data should instead be treated as non-Euclidean data [15]. Therefore, to overcome this limitation, Graph Convolution Network (GCN) that can more efficiently analyse non-Euclidean structured data has been introduced into solving the traffic forecasting problem on large-scale traffic networks. It conducts convolutional operations on non-Euclidean data for obtaining the relationships of traffic data in space domain. For example, [16] built the STGCN model consisting of two spatial-temporal convolutional blocks (named ST-Conv blocks) and a fully-connected layer for traffic forecasting on road network-wide. [17] proposed the TGC-LSTM model that uses GCN to capture the spatial dependencies on the road network-wide and utilises LSTM to extract temporal dependencies while Tang [18] jointed the attention mechanism into GCN and then developed a GAGCN model to forecast traffic flow.

Based on the most recent literature, in this paper, a novel deep learning framework is to be developed to resolve the long-term traffic forecasting problem, aiming to provide more accurate forecasting on large-scale road networks. The proposed framework takes the advantages of Graph Convolution Network and Transformer on the different spatial dependency analysis and the Sequence-to-Sequence architecture on the temporal dependency analysis, which is denoted as GCNT-Seq2Seq. The traffic forecasting problem will firstly be described in Section II and details of the proposed model will be explained in Section III. The proposed model will be evaluated against other existing forecasting models based on real world traffic dataset in Section IV. Finally, this work will be concluded in Section V.

II. Problem Formulation

The traffic forecasting problem in this paper is defined on a large-scale road network with N detectors. Therefore, the road graph used in the proposed model has N nodes and is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes with $|\mathcal{V}| = N$. $\mathcal{E}$ is the set of edges representing physical connectivity between detectors. Generally, $\mathcal{G}$ can be represented by $A \in \mathbb{R}^{N\times N}$, the $N \times N$ symmetric adjacency matrix, with its element $A_{i,j} = 1$ if there exists a link between node i and j and 0 otherwise. Considering that the future traffic states of a detector are influenced by its own historical states, the road graph, $\mathcal{G}$, is represented by $\tilde{A} = (A + I_N) \in \mathbb{R}^{N\times N}$ where I_N is the $N \times N$ identity matrix. $\tilde{A}$ only describes the connectivity of neighbors one hop away from each node (i.e., $1 - hop$ neighborhood). Due to considering that traffic speed propagates downstream while traffic congestion propagates upstream and the dataset used in this work includes traffic congestion, we introduce the notion of $k - hop$ neighborhood to represent the set of nodes that are reachable within k hops from the targeted node and define the $k - hop$ neighborhood matrix as $\tilde{A}^k \in \mathbb{R}^{N\times N}$.

The traffic data from the road network with N detectors is described as $x_t = \{x_t^1, x_t^2, \ldots, x_t^i, \ldots, x_t^{N-1}, x_t^N\}; x_t \in \mathbb{R}^N, (i = 1, 2, 3, \ldots, N)$, and x_t^i denotes the traffic data measured at detector i at t^{th} time step. Typically, a time step can represent 5, 15, 30, 45 and 60 minutes [19]. In this paper, the dataset used for evaluating the proposed model considers 5 minutes as a time step. Then $X \in \mathbb{R}^{T\times N}$ (cf. Eq. (1)) represents traffic data collected from N detectors in the network for T previous time steps. Conversely, the traffic data for the future is written as $X' = \{x_{t+1}, x_{t+2}, \ldots, x_{t+T'}\} \in \mathbb{R}^{T'\times N}$ where T' is the forecasting horizon. Generally, traffic forecasting problems can be categorized into short- ($T' < 30$ minutes) and long-term ($T' \geq 30$ minutes). Since we aim to solve the long-term traffic forecasting problem on large-scale road network, this paper covers timescales $T' = \{6, 12, 18, 24\}$ corresponding to $\{30, 60, 90, 120\}$ minutes.

$$\begin{aligned} X = \{x_{t-T+1}, x_{t-T+2}, \ldots, x_{t-1}, x_t\}; \\ X \in \mathbb{R}^{T\times N}, T = 1, 2, 3, \ldots \end{aligned} \tag{1}$$

Based on traffic data and the road graph described above, the traffic forecasting problem for the proposed approach in this paper can be formulated as Eq. (2).

$$\tilde{X}' = \boldsymbol{F}\Big(X; \mathcal{G}(\mathcal{V}, \mathcal{E}, \tilde{A}^k)\Big) \tag{2}$$

where the objective is to learn the mapping function $F(.)$ and compute the traffic data in the next T' time steps based on the historical traffic data in T previous time steps and the road graph $\mathcal{G}$. For fitting the training phase, both input X and targeted X' need to be formatted to $X \in \mathbb{R}^{B\times T\times N}$ and $X' \in \mathbb{R}^{B\times T'\times N}$ where B is the batch size.

III. A Novel Proposed Framework

Fig. 1 presents the framework of the proposed deep learning model, named GCNT-Seq2Seq, that is designed to comprehensively analyse the spatial and temporal dependencies of traffic data. GCNT-Seq2Seq is developed under the sequence-to-sequence architecture consisting of an encoder and a decoder for the long-term dependency analysis. The encoder is used to learn the historical information and encodes it to a context vector. The decoder is utilised to decode the context vector to the final forecasting. The modules of the sequence-to-sequence architecture is built based on graph convolution neural networks and the transformer for analysing spatial dependencies from original traffic data and the road graph data. The following content will explain the GCNT-Seq2Seq model on the spatial dependency analysis and the temporal dependency analysis in detail, respectively.

GCNT as the embedded module of the sequence-to-sequence architecture is used to analyse the spatial dependencies. Fig. 2 displays the GCNT module that consists of L_g graph convolutional neural (GCN) layers and L_s transformer layers in parallel. Considering the ${l_g}^{th}$ GCN layer at the

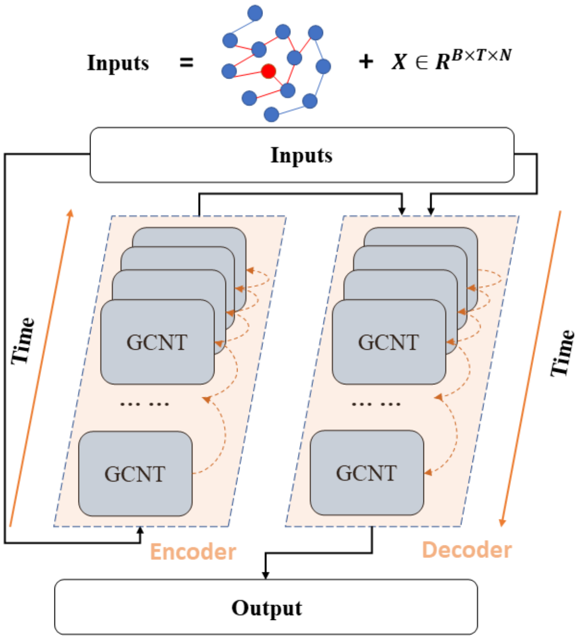


Fig. 1: The framework of GCNT-Seq2Seq.

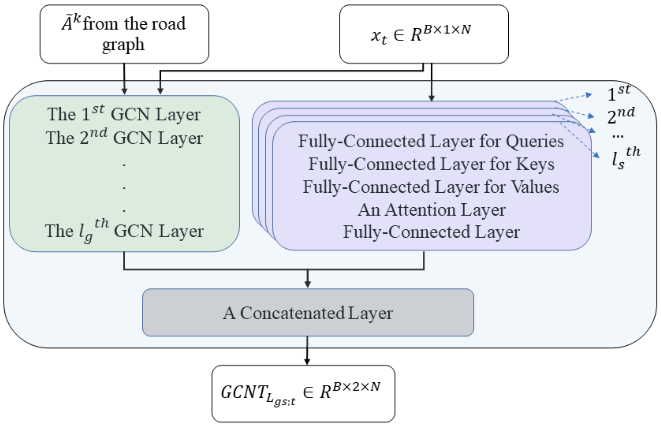

$\tilde{A}^k$ from the road graph
$x_t \in R^{B\times 1\times N}$
The 1^{st} GCN Layer
The 2^{nd} GCN Layer
.
.
.
The $l_g{}^{th}$ GCN Layer
Fully-Connected Layer for Queries
Fully-Connected Layer for Keys
Fully-Connected Layer for Values
An Attention Layer
Fully-Connected Layer
1^{st}
2^{nd}
…
$l_s{}^{th}$
A Concatenated Layer
$GCNT_{L_{gs:t}} \in R^{B\times 2\times N}$

Fig. 2: The module of GCNT

t^{th} time step as an example, the convolution operation is conducted on the $k-hop$ neighbourhood matrix $\tilde{A}^k$ joined to the output of the $(l_g-1)^{th}$ GCN layer for obtaining spatial features using Eq. (3). Due to $\tilde{A}^k$ describing connections of detectors in the $k-hop$ neighbourhood, obtained spatial features from the GCN layers can be considered as local spatial features.

$$GCN_{l_g;t} = ReLU\Big((W_{l_g;t} * \tilde{A}^k)GCN_{(l_g-1);t}\Big) + GCN_{(l_g-1);t} \tag{3}$$

where $GCN_{(l_g-1);t} \in \mathbb{R}^{B\times N}$ is the output of the $(l_g-1)^{th}$ GCN layer and is treated as the input of the $l_g{}^{th}$ GCN layer. $GCN_{l_g;t} \in \mathbb{R}^{B\times N}$ is the output of the $l_g{}^{th}$ GCN layer and the input of the 1^{st} GCN layer is $x_t \in \mathbb{R}^{B\times N}$. $W_{l_g;t} \in \mathbb{R}^{N\times N}$ is the weight matrix of $\tilde{A}^k$ and $ReLU$ is the activation function of GCN layers.

Meanwhile, transformer layers analyse spatial dependencies from the other aspect by the attention mechanism to assign individual weights to neighbours of the targeted detector so as to contribute to the targeted detector. Each transformer layer consists of three fully-connected layers, an attention layer and a linear layer. Considering the $l_s{}^{th}$ transformer layer at the t^{th} time step as an example, three fully-connected layers are used to generate the multi-head inputs of queries $Q_{l_s;t} \in \mathbb{R}^{B\times(H\times d_q)\times N}$, keys $K_{l_s;t} \in \mathbb{R}^{B\times(H\times d_k)\times N}$ and values $V_{l_s;t} \in \mathbb{R}^{B\times(H\times d_v)\times N}$ by Eq. (4), in which H is the number of multi-heads and d_q, d_k and d_v are the number of embedded features for $Q_{l_s;t}$, $K_{l_s;t}$ and $V_{l_s;t}$, respectively. The spatial weight matrix is generated by multiplying the transposition of $K_{l_s;t}$ by $Q_{l_s;t}$ and then utilised to obtain spatial features using Eq. (5). Due to the spatial weight matrix $\frac{Q_{l_s;t}K_{l_s;t}{}^{\mathsf{T}}}{\sqrt{d_k}} \in \mathbb{R}^{B\times H\times N\times N}$ enabling all other detectors to have individual weights for the targeted detector, spatial features obtained here are considered as global spatial features.

$$\begin{aligned} Q_{l_s;t} &= W^q_{l_s;t} S_{(l_s-1);t}{}^{\mathsf{T}} \\ K_{l_s;t} &= W^k_{l_s;t} S_{(l_s-1);t}{}^{\mathsf{T}} \\ V_{l_s;t} &= W^v_{l_s;t} S_{(l_s-1);t}{}^{\mathsf{T}} \end{aligned} \tag{4}$$

$$S_{l_s;t} = ReLU\left(W^s_{l_s;t}\Big(softmax(\frac{Q_{l_s;t}K_{l_s;t}{}^{\mathsf{T}}}{\sqrt{d_k}})V_{l_s;t}\Big) + b^s_{l_s;t}\right) + S_{(l_s-1);t} \tag{5}$$

where $S_{(l_s-1);t}{}^{\mathsf{T}} \in \mathbb{R}^{B\times N}$ is the transposition of the output of the $(l_s-1)^{th}$ transformer layer as the input of the $l_s{}^{th}$ transformer layer. S_{l_s} is the output of the $l_s{}^{th}$ transformer layer and the input of the 1^{st} transformer layer is x_t. $W^q_{l_s;t}$, $W^k_{l_s;t}$ and $W^v_{l_s;t}$ are weight matrices of queries, keys and values, respectively. $W^s_{l_s;t}$ is the weight matrix of the linear layer and $b^s_{l_s;t}$ is the related bias. $ReLU$ is an activation function.

Finally, the output of the last GCN layer, $GCN_{L_g;t}$, and the output of the last transformer layer, $S_{L_s;t}$, are concatenated, and then pass a linear layer to generate local-global spatial features $GCNT_{L_{gs};t}$ using Eq. (6). In addition, the residual connection network [20] is used in each GCN and transformer layer to ensure the stable training and also supplement the important information hidden in negative values that are neglected by the $ReLU$ activation function.

$$GCNT_{L_{gs};t} = W_{L_{gs};t}concat(GCN_{L_g;t}, S_{L_s;t}) + b_{L_{gs};t} \tag{6}$$

where $W_{L_{gs};t}$ and $b_{L_{gs};t}$ are the weight matrix and the bias, respectively.

Seq2Seq consists of LSTMs as the encoder and the decoder, which is used to embed our GCNT module for extracting and delivering temporal features from spatially-fused features. Taking the t^{th} time step as an example, the encoder takes the $GCNT_{L_{gs};t}$ as the input and encodes the spatially-fused features using Eq. (7).

$$\begin{aligned} f_t &= \sigma_g(w_f \cdot GCNT_{L_{gs};t} + u_f \cdot h_{t-1} + b_f) \\ i_t &= \sigma_g(w_i \cdot GCNT_{L_{gs};t} + u_i \cdot h_{t-1} + b_i) \\ o_t &= \sigma_g(w_o \cdot GCNT_{L_{gs};t} + u_o \cdot h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(w_c \cdot GCNT_{L_{gs};t} + u_c \cdot h_{t-1} + b_c) \\ h_t &= o_t \circ \sigma_h \times (c_t) \end{aligned} \tag{7}$$

where w_f, w_i, w_o and w_c are the weights of the forget gate f_t, the input gate i_t, the output gate o_t and the cell state c_t respectively while b_f, b_i, b_o and b_c are the corresponding biases for each gate and the cell state. Furthermore, u_f, u_i, u_o and u_c are the weights of the last hidden state h_{t-1}. σ_g denotes a sigmoid function (= $\frac{1}{1+e^{-x}}$) in three gates and the operator $\circ$ denotes Hadamard product. σ_c and σ_h are hyperbolic tangent function ($tanh(x)$) for the cell state and the final output. h_t is considered as the output of the encoder and carries historical information. In the decoder, the output of the encoder h_t is treated as the initialised hidden state and $GCNT_{L_{gs};t}$ is still considered as the input. The output of the decoder is the final forecasting.

IV. Experiments

A. Data Description

Dataset used for evaluating the proposed model and comparing our model against well-known existing models is collected from the real-world road network, named `METR-LA` [21]. Fig. 3 displays the locations of loop detectors in `METR-LA`. It includes 207 detectors and covers 4 months of traffic speed data from 1^{st} of March to 30^{th} of June in 2012. A time step is 5 minutes and the number of observed traffic data points is 7,094,304 (= 34272 × 207). This dataset misses some data points so that it brings more challenges. To evaluate and valid the robustness of the proposed model, discontinuous missing data points are replaced by the mean of the last and next traffic data and continuous missing data points are set as zeros to simulate traffic incidents such as traffic accidents and congestion. In addition, an undirected road graph with edge weights is used to construct the neighbourhood matrix. The pairwise road distances between detectors are first computed and then a thresholded Gaussian Kernel in [22] is utilised to build the weighted matrix.

B. Performance Metrics

To measure the results, three types of metrics named Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root-Mean Square Error (RMSE) in the literature [14][17] are used and computed as Eq. (8), Eq. (9) and Eq. (10).

$$MAE = \frac{1}{N \times T'} \cdot \sum_{i=1}^{N} \sum_{t=1}^{T'} |x_t^i - \tilde{x}_t^i| \tag{8}$$

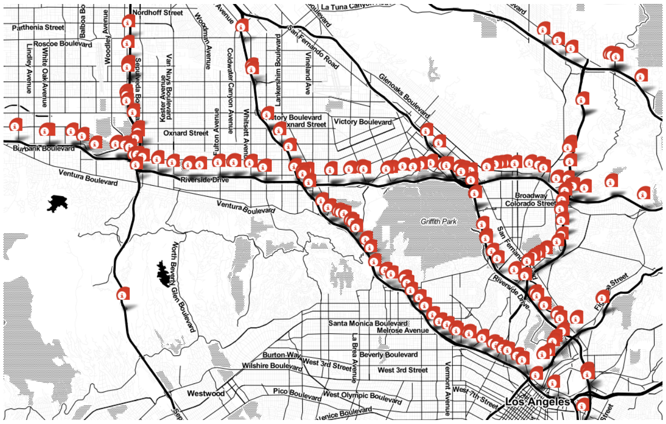


Fig. 3: Locations of loop detectors in `METR-LA`

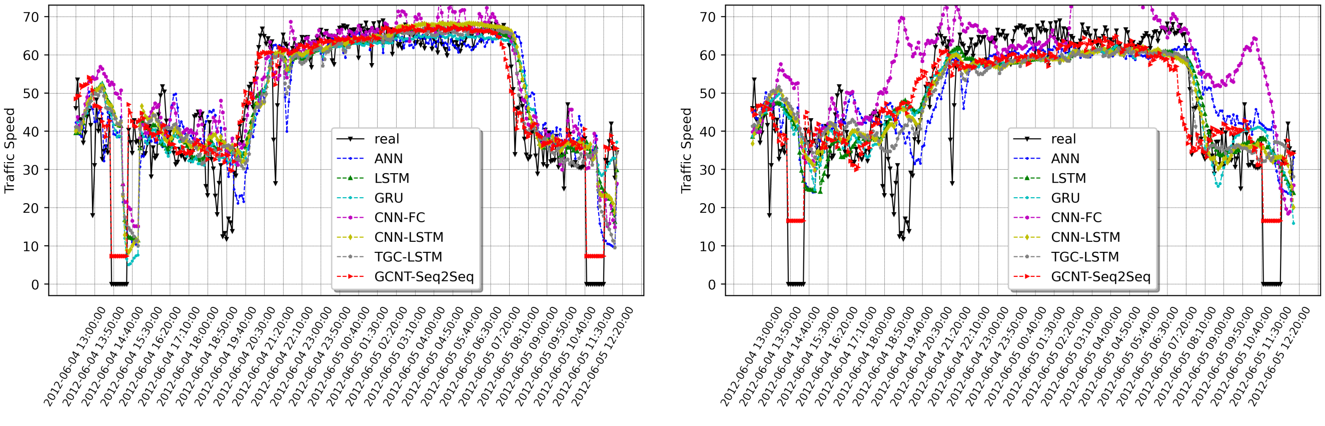
$$MAPE = \frac{1}{N \times T'} \cdot \sum_{i=1}^{N} \sum_{t=1}^{T'} \frac{|x_t^i - \tilde{x}_t^i|}{x_t^i} \times 100\% \tag{9}$$

$$RMSE = \left[\frac{1}{N \times T'} \cdot \sum_{i=1}^{N} \sum_{t=1}^{T'} (x_t^i - \tilde{x}_t^i)^2 \right]^{\frac{1}{2}} \tag{10}$$

where MAE presents the average absolute difference between the forecasted and real traffic speed. It is used to measure absolute forecasting error. MAPE is the relative difference between the forecasted and real traffic speed and is utilised to measure relative forecasting error. RMSE is the standard deviation of the residuals where residual is the difference between forecasted and real traffic speed.

C. Parameter Settings

To optimise the GCNT-Seq2Seq model and obtain high forecasting accuracy, there are several hybrid-parameters that need to be tuned including historical time steps T, targeted time steps T', the number of multi-heads H, the number of embedded features $\{d_q, d_k, d_v\}$, the learning rate r, batch size B and the number of epochs. Based on the well-known existing works, the historical time steps T, targeted time steps T', the number of multi-heads H, the number of embedded features $\{d_q, d_k, d_v\}$, and batch size B are set as 12, 24, 8, $\{8, 8, 8\}$ and 32, respectively. For the learning rate setting, generally, a too small learning rate will make a training algorithm converge slowly while a too large learning rate will make the algorithm diverge. Therefore, finding an optimal learning rate is very important to improve the performance of the algorithm. However, the experimental method to find the best learning rate is time-consuming. In our work, the method named Cyclical Learning Rates (CLR) in [23] is used to optimise the learning rate as $1.02e^{-03}$. In addition, *stop early* strategy is used to optimise the number of epochs and avoid the problem of over-fitting. It means the training process will be stopped when the training loss continues to decrease in 10 epochs while the validation loss increases. Finally, we follow the convention and use 70% of the dataset for training, 10% for validation and 20% for testing.



(a)

(b)

Fig. 4: (Color Online) Real (black color) and forecasted (other colors) traffic speed $(miles/h)$ in a day with 288 $(= \frac{1days*24hours*60minutes}{5minutes})$ time intervals from all models on `METR-LA` with a time interval = 5 minutes. The x-axis represents the time and the y-axis is traffic speed (miles/hour). The forecasting horizons are 30 minutes in (a) and 120 minutes in (b), respectively. The red lines in (a) and (b) represent forecasted traffic speed from our GCNT-Seq2Seq

TABLE I: Experimental results from all models on `METR-LA`

Model Name	METR-LA		
	MAE(T'=6/12/18/24)	MAPE(T'=6/12/18/24)	RMSE(T'=6/12/18/24)
ANN	5.0232 / 6.7519 / 8.1165 / 9.1839	13.12% / 18.20% / 21.68% / 24.42%	9.8455 / 11.8983 / 13.1064 / 13.8129
LSTM	5.2023 / 6.6489 / 7.6383 / 8.2251	12.74% / 16.44% / 18.68% / 21.21%	9.4332 / 11.1926 / 12.2611 / 12.8663
GRU	5.0715 / 6.2108 / 7.0753 / 7.7387	12.67% / 15.64% / 18.17% / 20.03%	**9.1238 / 10.6058** / 11.7414 / 12.4671
CNN-FC	6.4447 / 8.0232 / 8.4805 / 9.4339	14.77% / 18.40% / 19.91% / 21.88%	9.7793 / 11.5983 / 12.0401 / 13.0291
CNN-LSTM	5.2192 / 6.4425 / 7.5352 / 8.3240	12.70% / 16.03% / 18.79% / 20.96%	9.3857 / 10.9365 / 12.1356 / 12.8291
TGC-LSTM	5.2822 / 6.6194 / 7.6074 / 8.2736	12.62% / 15.99% / 18.64% / 20.54%	9.5018 / 11.1085 / 12.1906 / 12.8627
GCNT-Seq2Seq	**4.9956 / 6.1774 / 6.9358 / 7.5813**	**12.10% / 15.38% / 17.44% / 19.41%**	9.2731 / 10.8090 / **11.6876 / 12.3108**

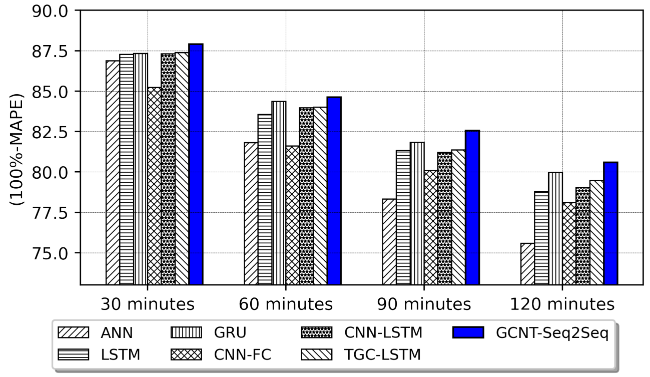


Fig. 5: The forecasting accuracy (100%-MAPE) from all comparison models.

D. *Results and Discussion*

To evaluate the proposed model, six baselines are used for comparison experiments, including 1) one linear feature-based model, ANN [24]; 2) two temporal feature-based models, LSTM [25] and GRU [26]; 3) one spatial feature-based model, CNN-FC [27]; 4) two spatial and temporal feature-based models, CNN-LSTM [14] and TGC-LSTM [17].

Fig. 4 presents the real (black color) and forecasted (other colors) traffic speed data in a day from the GCNT-Seq2Seq model and other six baselines. Fig. 4 (a) and (b) consider 30 minutes and 120 minutes as forecasting horizons, respectively. Overall, GCNT-SeqSeq can efficiently forecast the trend of traffic speed changes, even for longer forecasting horizons. Besides, GCNT-Seq2Seq also capture the sudden changes caused by mission data or traffic incidents.

TABLE I presents experimental results from all competition models for different forecasting horizons. $T' = \{6, 12, 18, 24\}$ are responding to consider $\{30, 60, 90, 120\}$ minutes as forecasting horizons. It is observed clearly that the proposed model, GCNT-Seq2Seq, achieves the best performance among all competition models. Its MAEs, MAPEs and RMSEs for all forecasting horizons are almost the lowest. It means the forecasting accuracy of GCNT-Seq2Seq is the highest, which can be also observed in Fig. 5 that displays the forecasting accuracy (100%-MAPE) of all models. From Fig. 5, the differences of forecasting accuracy between GCNT-Seq2Seq and other models become more obvious over longer forecasting horizons. It means the superiority of our model is more obvious for longer forecasting horizons. Even for 120 minutes as forecasting horizon, the forecasting accuracy of GCNT-Seq2Seq is still more than 80% and it is also the only one model achieving over 80% accuracy among all comparison models.

From TABLE I, the linear feature-based model, ANN, that relies on the linear relationship of traffic speed data in time domain for forecasting almost obtains the worst results excluding 30 minutes as the forecasting horizon where the spatial feature-based model, CNN-FC, is the worst one. The probable reason is that linear relationships for longer forecasting horizons are not obvious while the spatial dependencies among road network become more important. Therefore, CNN-FC model that mainly focuses on spatial feature extraction by the convolution kernels performs better than ANN for longer forecasting horizons. Between two temporal feature-based models, LSTM and GRU, GRU always outperforms LSTM for all different forecasting horizons. This conclusion was confirmed in [12] by numerous experiments. Both CNN-LSTM and TGC-LSTM achieve similar results, because they are able to analyse both spatial and temporal features for final forecasting.

V. Conclusion

In this paper, the problem of forecasting long-term traffic on large-scale road network is addressed and a novel deep learning framework, named GCNT-Seq2Seq, is developed based on the most advanced technologies. The proposed framework takes the advantages of Graph Convolution Network (GCN) and Transformer on the different spatial dependency analysis and the advantage of the Sequence-to-Sequence (Seq2Seq) architecture on the temporal dependency analysis. GCN is used to extract the local spatial features by operating convolution on the $k - hop$ neighbourhood matrix and Transformer is utilised to capture the global spatial features by assigning individual weights to neighbours of the targeted detector so as to contribute to the targeted detector. The concatenation of local and global spatial features is embedded into the Seq2Seq architecture for temporal feature extraction and final forecasting. The proposed framework is compared to the existing well-known models using the real world dataset with missing data and frequent traffic incidents. Among all the models being compared, our proposed model shows the best performance and can even achieve more than 80% accuracy for forecasting two hours traffic status in advance.

References

[1] L. Figueiredo, I. Jesus, and et.al., "Towards the development of intelligent transportation systems," in IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585), 2001, pp. 1206–1211.

[2] Y. Lin, P. Wang, and M. Ma, "Intelligent transportation system (its): Concept, challenge and opportunity," in IEEE 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids), 2017, pp. 167–172.

[3] M. S. Ahmed and A. R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques. Transportation Research Board, 1979, no. 722.

[4] S. Ho and M. Xie, "The use of arima models for reliability forecasting and analysis," Computers & industrial engineering, vol. 35, no. 1-2, pp. 213–216, 1998.

[5] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," Transportation Research Record, vol. 1644, no. 1, pp. 132–141, 1998.

[6] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," Neurocomputing, vol. 50, pp. 159–175, 2003.

[7] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 2, pp. 871–882, 2013.

[8] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," Transp. Res. Part C Emerg. Technol., vol. 66, pp. 61–78, 2016.

[9] W. Huang, G. Song, and et.al., "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2191–2201, 2014.

[10] Y. Lv, Y. Duan, and et.al., "Traffic flow prediction with big data: a deep learning approach," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 865–873, 2014.

[11] W. Zhang, Y. Yu, and et.al., "Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning," Transportmetrica A: Transport Science, vol. 15, no. 2, pp. 1688–1711, 2019.

[12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in NIPS Workshop on Deep Learning, 2014.

[13] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," arXiv preprint arXiv:1612.01022, 2016.

[14] Y. Liu, H. Zheng, and et.al., "Short-term traffic flow prediction with conv-lstm," in IEEE 9th International Conference on Wireless Communications and Signal Processing (WCSP), 2017, pp. 1–6.

[15] E. Ahmed, A. Saint, A. E. R. Shabayek, and et.al., "A survey on deep learning advances on different 3d data representations," arXiv preprint arXiv:1808.01462, 2018.

[16] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3634–3640.

[17] Z. Cui, K. Henrickson, and et.al., "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," IEEE Transactions on Intelligent Transportation Systems, 2019.

[18] C. Tang, J. Sun, Y. Sun, and et.al., "A general traffic flow prediction approach based on spatial-temporal graph attention," IEEE Access, vol. 8, pp. 153 731–153 741, 2020.

[19] P. J. Bickel, C. Chen, J. Kwon, and et.al., "Measuring traffic," Statistical Science, pp. 581–597, 2007.

[20] K. He, X. Zhang, S. Ren, and et.al., "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[21] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in International Conference on Learning Representations, 2018.

[22] D. I. Shuman, S. K. Narang, P. Frossard, and et.al., "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," IEEE signal processing magazine, vol. 30, no. 3, pp. 83–98, 2013.

[23] L. N. Smith, "Cyclical learning rates for training neural networks," in IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 464–472.

[24] A. Csikós, Z. J. Viharos, K. B. Kis, and et.al., "Traffic speed prediction method for urban networks—an ann approach," in International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). IEEE, 2015, pp. 102–108.

[25] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with lstm recurrent neural network," in IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 1–6.

[26] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in IEEE 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2016, pp. 324–328.

[27] X. Ma, Z. Dai, Z. He, and et.al., "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," Sensors, vol. 17, no. 4, p. 818, 2017.