Transportation Research Part C 108 (2019) 12–28

Contents lists available at ScienceDirect

Transportation Research Part C

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

An effective spatial-temporal attention based neural network for traffic flow prediction

Loan N.N. Do[a,*], Hai L. Vu[b], Bao Q. Vo[a], Zhiyuan Liu[c], Dinh Phung[d]

[a] Faculty of Science, Engineering and Technology, Swinburne University of Technology, Australia
[b] Department of Civil Engineering, Monash University, Australia
[c] Jiangsu Key Laboratory of Urban ITS, Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Southeast University, China
[d] Faculty of Information Technology, Monash University, Australia

ARTICLE INFO

Keywords:
Traffic flow prediction
Traffic flow forecasting
Deep learning
Neural network
Attention

ABSTRACT

Due to its importance in Intelligent Transport Systems (ITS), traffic flow prediction has been the focus of many studies in the last few decades. Existing traffic flow prediction models mainly extract static spatial-temporal correlations, although these correlations are known to be dynamic in traffic networks. Attention-based models have emerged in recent years, mostly in the field of natural language processing, and have resulted in major progresses in terms of both accuracy and interpretability. This inspires us to introduce the application of attentions for traffic flow prediction. In this study, a deep learning based traffic flow predictor with spatial and temporal attentions (STANN) is proposed. The spatial and temporal attentions are used to exploit the spatial dependencies between road segments and temporal dependencies between time steps respectively. Experiment results with a real-world traffic dataset demonstrate the superior performance of the proposed model. The results also show that the utilization of multiple data resolutions could help improve prediction accuracy. Furthermore, the proposed model is demonstrated to have potential for improving the understanding of spatial-temporal correlations in a traffic network.

1. Introduction

Intelligent transports systems (ITS) have emerged to tackle traffic management problems due to the increasing number of urban vehicles. Typically, the goal of ITS is to facilitate effective urban planning, route guidance, solving traffic congestion, and other applications in transportation. Accurate and timely traffic flow prediction has been a key problem in ITS research, which has led to an intensive body of work in recent years. Traffic flow prediction can be defined as the forecasting of traffic flow given the values of observed or historical traffic data. Prediction within one or several hours ahead is typically called short-term prediction (Vlahogianni et al., 2014; Ermagun and Levinson, 2018). This task is made challenging by nonlinear spatial-temporal dynamics in traffic. These dynamic dependencies are caused by the changes in traffic demands and capacities from time to time. Understanding the importance of these characteristics in traffic forecasting, many approaches have been proposed to exploit these dependencies while making predictions.

Traffic flow forecasting is typically done using statistical or machine learning approaches. Different types of methods can also be

* Corresponding author.
E-mail addresses: ndo@swin.edu.au (L.N.N. Do), hai.vu@monash.edu (H.L. Vu), bvo@swin.edu.au (B.Q. Vo), zhiyuanl@seu.edu.cn (Z. Liu), dinh.phung@monash.edu (D. Phung).

https://doi.org/10.1016/j.trc.2019.09.008
Received 2 February 2019; Received in revised form 12 August 2019; Accepted 10 September 2019
Available online 17 September 2019

combined to perform this task, such as applying clustering techniques to classify data before making predictions, or jointly considering different prediction models (Chen et al., 2001; Vlahogianni et al., 2004; Vlahogianni, 2015). One of the most popular statistical approaches is to use Auto-regressive integrated moving average (ARIMA) models. ARIMA and its variants have proved useful in many cases (Van Der Voort et al., 1996; Williams, 2001; Williams and Hoel, 2003). In these traffic estimations, spatial weight matrices and network weight matrices have been employed (Guan et al., 2018; Ermagun and Levinson, 2019). To explore spatial-temporal correlations and perform similar traffic state estimations, a stochastic cell transmission model was also proposed to be used with a multivariate best linear predictor (Pan et al., 2013). However, these models are linear time-series models and are not capable of dealing with the nonlinearities in traffic flow dependencies. Machine learning approaches such as neural networks (NN) and support vector regressions (SVR) have been proposed to capture nonlinear spatial-temporal relationships. In this branch of research, NN-based models, including deep learning models, have been shown to be particularly promising for traffic state prediction, both short-term and long-term, through recent studies (Lv et al., 2015; Polson and Sokolov, 2017; Vlahogianni et al., 2005; Laña et al., 2019). Deep learning models can now be trained to capture more complex traffic correlations due to advancements in computational power. As a result, many traffic predictors employing Convolutional Neural Networks (CNN) or combined with Recurrent Neural Networks (RNN) have been proposed to mine both spatial and temporal features when making predictions (Ma, Yu et al., 2015; Fouladgar et al., 2017; Ma et al., 2017; Yu et al., 2017). Although these models achieved encouraging prediction accuracy, most of them can only be applied for traffic networks as simple as a sequence of road segments, whereas urban network topologies are generally much more complicated. In addition to this problem, the existing models tend to extract static spatial-temporal dependencies in traffic, whereas these dependencies evolve over time. Another issue is that most existing deep learning traffic flow predictors lack the interpretability due to the “black-box” nature of this type of model (Karlaftis and Vlahogianni, 2011; Zhang et al., 2014; Do et al., 2019). A better understanding of the spatial-temporal correlations in a traffic network exploited by the models would be helpful for traffic planning and management.

To address these issues, we propose a novel deep learning architecture, Spatial-Temporal Attention based Neural Network (STANN), for traffic flow prediction. STANN is a sequence to sequence model which tackles spatial-temporal dependencies through the use of convolutional Gated Recurrent Unit (GRU) components and the attention mechanism. The attention mechanism has been shown as a potential approach for improving both deep neural network accuracy and interpretability (Barbieri et al., 2018; Wu et al., 2018). To the best of our knowledge, this is one of the first attempts to apply the attention mechanism in traffic flow prediction over both time and space dimensions. To facilitate the training and testing of the proposed model, as well as its performance evaluation, traffic loop detector data collected from the SCATS (Sydney Coordinated Adaptive Traffic System) in Melbourne, Australia will be utilized in this paper. The dataset contains traffic volume (or traffic count) and the degree of saturation, which represents the density of traffic flow along a traffic lane and is the ratio of the effectively used green time to the total available green time, of every road segment per minute. The comparisons with the state-of-the-art prediction methods show the effectiveness of the proposed method in traffic flow prediction. In addition, the analyses of the result illustrate the proposed model’s capability in providing a certain level of understanding about spatial-temporal traffic correlations.

The main contributions of this study are summarized as follows:

- We propose a method to construct spatial attentional matrices, which contain attention weights representing the relationships between road segments in a traffic network.
- We propose a traffic flow prediction model with spatial-temporal attention (STANN), which exploits both temporal and spatial correlations in traffic using the attention mechanism.
- We propose the use of different data resolutions to further improve the proposed model’s performance.
- We analyze the attention scores obtained from the proposed model. The analyses uncover interesting spatial-temporal traffic correlations and provide a posterior interpretation of relations among sensors across the space and time dimensions.
- We conduct comprehensive performance comparisons using a real-word dataset and the proposed model demonstrates its superiority over state-of-the-art baseline methods.

The paper is organized as follows: Section 2 gives a summary of state-of-the-art research in exploring correlations in traffic to predict traffic flow. We then propose the STANN model in Section 3. Section 4 discusses the experiment design and the performances of the tested models. Conclusions are then drawn in Section 5.

2. Related work

2.1. Spatial-temporal NN-based traffic flow prediction models

Temporal and spatial correlations in traffic data have been explored using different methods to be utilized in neural network based traffic prediction models.

Temporal dependencies are the most common features used to develop traffic predictors. Most of the existing neural network models use preceding values and/or values at previous days or weeks as inputs to predict future values. Such prediction models include Smith and Demetsky (1994), Chen et al. (2001), Leng et al. (2013) and Jia et al. (2016).

Various types of neural networks dealing with sequences have also been proposed for traffic flow prediction problems such as Time-delay Neural Networks (TDNNs) (Lingras and Mountford, 2001), Long-short Term Memory Neural Networks (LSTM NNs) (Ma, Tao et al., 2015; Wu and Tan, 2016; Jia et al., 2017; Zhao et al., 2017), Gate Recurrent Unit Neural Networks (GRU NNs) (Fu et al.,

2016), and other Recurrent Neural Networks (RNNs) (Zhang, 2000; Ishak et al., 2003). The structures of these neural network models incorporate time dependency naturally using sequences of inputs and continuous feedback between time steps. These models demonstrated superior performance compared with existing traffic flow prediction methods. Among those architectures, LSTM NNs and GRU NNs have become favorite choices in recent models due to their capability for dealing with long sequences. As GRU is a variant of LSTM, both models have comparable performances. However, GRU networks contain fewer parameters and so are faster to train. In Fu et al. (2016), both LSTM and GRU outperformed ARIMA, and the performance of the GRU was slightly better than the LSTM. The GRU NN achieved the mean absolute error (MAE) lower than roughly 10% and 5% on average compared with ARIMA and LSTM NN respectively.

As well as temporal dependencies, spatial dependencies have also been exploited in various research in this field. The utilization of spatial dependencies can be as simple as including data from all the links in the network or data from upstream and downstream links as inputs (Lv et al., 2015; Poslon and Sokolov, 2017; Ishak et al., 2003; Zheng et al., 2006; Zhu et al., 2014). The latter method was shown to be more helpful as not all the links in the network are related to each other. The spatial-temporal features were also explored more sophisticatedly by constructing useful inputs for traffic flow predictors through the extraction of correlations in the data (He et al., 2008; Sun et al., 2012; Zhao et al., 2017); considering each location as a module in a modular network (Vlahogianni et al., 2007); considering each location as a task in a multitask DBN model (Huang et al., 2014).

Utilizing Convolutional Neural Network (CNN) based models, Ma et al. (2017) transformed a traffic network to greyscale images to predict traffic speed. Two-dimensional space-time matrices were used to represent the temporal and spatial correlations and 2D-convolution operations were applied. The number of columns was the number of prior time intervals used for prediction, and the number of rows was the number of road segments. In this way, each cell of the matrix stored the traffic speed of a road segment at a specific time. Compared to traditional NNs and other baselines, the proposed model showed an improvement in average accuracy with 27.96%. Fouladgar et al. (2017) also transformed the traffic network data into space-time matrices and applied 2D convolutions. However, unlike the above study, a matrix was constructed for each target location in this work. Each matrix included historical data of the in-flow sequence of locations, the target location and the out-flow sequence of locations. Similarly, in Wu and Tan's model, traffic states of a sequence of segments were transformed into vectors (Wu and Tan, 2016). The CNN took these vectors as inputs and applied 1D-convolution operations to mine spatial features. In this study, two LSTMs were also used in parallel to extract short-term temporal features and periodic features. All these spatial and temporal features were then concatenated to a feature vector which was used to make predictions through a regression layer. It is noticeable that the linear representation of a traffic network in the above model was only possible if the network structure was simple as a sequence of road segments. For more complex topologies, useful spatial features may not be extracted using this method.

To tackle this challenge, another work also employing the combination of CNN and LSTM was presented (Yu et al., 2017). In this work, a traffic network was converted to grid maps based on geographical coordinates. Traffic data was then mapped to the grids, so each grid represented a traffic state. These grids were used as inputs to the CNN to extract spatial features. These features were then processed through LSTM layers to exploit temporal correlations before going through a fully-connected layer to produce outputs. With a MAPE of 12.7%, the proposed model outperformed other deep learning models in the experiments, including LSTMs, CNNs and SAEs. However, the model's efficiency was a concern as the training and testing process was reported to be very time-consuming. Also in this line of research, a graph convolutional recurrent neural network was presented to deal with temporal and spatial traffic dynamics in complex traffic networks (Li et al., 2018). In this study, the sequence to sequence structure, which was originally proposed for neural machine translation (Sutskever et al., 2014), was applied. Both the encoder and decoder in this model were built from modified GRUs, which used convolution operations instead of traditional matrix multiplications. The convolution operations were based on the diffusion matrices constructed from the traffic network's topology. Cui et al. (2018) also proposed the use of a free-flow reachable matrix in a hybrid model of CNN and LSTM. This matrix determines if neighboring locations could influence each other based on the reachability between locations with free-flow speed. The reachable matrix is then used for the convolutions of data from neighboring road links. Deng et al. (2019) exploited the spatial-temporal correlations in a CNN by constructing input matrices for this CNN which include data of the most relevant locations in the previous steps. For each pair of locations, the relevance was defined by measuring the correlation in their historical values. Given the context of incomplete data, a subset of these locations with complete data was randomly selected for use. Another study also calculating the spatial correlations between locations based on their historical values was proposed by Dai et al. (2019). In this work, the locations were first placed in a square matrix in the order of their identifiers. The matrix was then rearranged to maximize the total spatial correlation score. Based on the positions of locations in this matrix, the input matrices for a CNN were constructed. For a similar task, a station-level bike-sharing demand prediction, the data-driven graph filter was used, instead of predefined adjacency matrices, in a graph CNN. This helps capture correlations between stations and the method was shown to be very promising (Lin et al., 2018) Although these models demonstrated their capabilities in dealing with more complex traffic networks and achieved promising performance, spatial-temporal correlations exploited were static though these correlations could be changed over time.

2.2. Attention mechanism

The attention mechanism has emerged as a method for helping NNs identify which parts of the input are more relevant and thus need more focus in a prediction task. This mechanism was first proposed to improve the performance of sequence to sequence models in neural machine translation (Bahdanau et al., 2014; Luong et al., 2015). It has been shown to be a promising approach in capturing the correlations between inputs and outputs while improving the interpretability of NN models. This method has also been applied in various areas such as image captioning, question answering systems, other NLP tasks, etc. (Rush et al., 2015; Xu et al., 2015; Bachrach

et al., 2017; Vaswani et al., 2017). The application of attention based models in traffic state prediction has still been very limited. Cheng et al. (2018) proposed an application of the attention mechanism over different orders of neighborhood road segments, which were called order slots. In this study, a hybrid structure of CNNs and LSTMs was applied to extract spatial-temporal features from different order slots of upstream and downstream locations for each target location. The model then measured how much attention each order slot required when making predictions for the target location. Another attempt to apply this mechanism to align traffic flows of previous days to the target day was also presented (Yao et al., 2018). The idea came from an observation of temporal shifting in traffic data over days. Using the attention mechanism to decide the importance of each considered time step of previous days to the predicted time of the target day was shown to be an effective treatment. At an early stage, these applications respectively demonstrated the attention mechanism's potential in tackling spatial and temporal dependencies in traffic data. This encourages the application of attention to traffic state prediction models in a more flexible and comprehensive way to deal with complex correlations between road segments which change over time in a traffic network.

Sharing a similar objective to the attention mechanism but using a different NN structure, Wu et al. (2018) proposed the use of an attention model in a traffic prediction model, which is called DNN-BTF. Unlike the above attention-based traffic predictors, this model considers attention over both spatial and temporal dimensions, so DNN-BTF could be a good representative of attention-based traffic prediction models. Therefore, the model is chosen to be one of the baselines in our experiments. Representing a traffic network as a sequence of locations, two-dimensional matrices with the speed values of all locations at all looked-back time steps were fed to the attention model. The attention model was a fully-connected feedforward NN generating a matrix with values from 0 to 1, which were called attention scores. The output matrix was then multiplied with a matrix of volumes of all locations at the previous time steps and the results were served as input for CNNs and GRUs. Although the model showed encouraging prediction accuracy, it works under an assumption that the traffic network could be linearly represented so it shares the same problem as other works mentioned above (Wu and Tan, 2016; Ma et al., 2017; Fouladgar et al., 2017). Additionally, the attention model constructed in this way implies that all the links in the network would influence each other, which is unlikely to be realistic, especially in a large network.

2.3. Deep learning models' interpretability

Though various deep learning models have been used for traffic state prediction with high accuracy, the interpretability of these models is still an open issue as the output of hidden layers are normally hard to interpret. It would be helpful to be able to identify the spatial and temporal components that influence the results of this type of model.

A limited number of studies have addressed this issue. Cui et al. (2018) proposed a traffic predictor combining traffic graph convolution and LSTM, which was shown to be capable of identifying the most influential road segments in road networks by visualizing weights. Another attempt has been made with the proposal of a path based deep learning framework for traffic speed prediction (Wang et al., 2019). In this work, the traffic network was divided into critical paths which were modeled using bidirectional LSTMs (Bi-LSTMs). The idea of using critical paths was shown to be helpful in providing more insights into the spatial-temporal correlations. By analyzing the output of the LSTM layers of the critical paths, the interpretability of the model was illustrated. In another approach, Wu et al. (2018) applied the paradigm of attention into a hybrid model comprised of CNN and RNN. It was shown that certain traffic knowledge learned from the traffic data could be extracted from their proposed attention-based model. Indeed, the attention mechanism helps make deep learning models more interpretable by assigning each element of the inputs an attention weight relating to its importance. The interpretability of such models has also been investigated and demonstrated in other tasks such as neural machine translation, image captioning, emoji prediction, etc. (Bahdanau et al., 2014; Xu, et al., 2015; Barbieri et al., 2018; Brown et al., 2018; Chen et al., 2019).

Motivated by the successful applications of models with attention, in this study, we propose a deep learning model with spatial-temporal attention (STANN) to predict traffic volumes. In this model, the idea of using attention was applied in both space and time dimensions to explore dynamic spatial-temporal correlations in traffic networks. In addition, by adopting the attention mechanism, the model is expected to benefit from the promising interpretability of this mechanism.

3. Methodology

3.1. Traffic prediction problem

A traffic network can be represented by a directed graph $G = (V, E)$ where V is the set of N nodes (road segments) and E is the set of M edges. $\forall\, v_i, v_j \in V$, if v_i and v_j are connected in the network with the direction from v_i to v_j, $(v_i, v_j) \in E$. This also means road segment v_i is an immediate upstream road segment of road segment v_j in the traffic network.

Let $Q^t \in \mathbb{R}^N$ be the traffic volume vector at time t. Similarly, $DS^t \in \mathbb{R}^N$ is the degree of saturation (DS) vector at time step t.

The problem can be defined as making traffic volume predictions $\widehat{Q}^{t+1}$, $\widehat{Q}^{t+2}$, $\cdots$,$\widehat{Q}^{t+p}$ (p is the prediction horizon) for each time step t, given traffic volumes and DS values of k previous time steps Q^t, Q^{t-1}, $\cdots$, Q^{t-k+1} and DS^t, DS^{t-1}, $\cdots$, DS^{t-k+1}.

3.2. Temporal attention

Sequence to sequence model has been shown to be a promising approach to deal with sequences (Sutskever et al., 2014; Cho et al., 2014; Luong et al., 2015; Chiu et al., 2018). This model takes a sequence of inputs and generates a sequence of prediction values. It

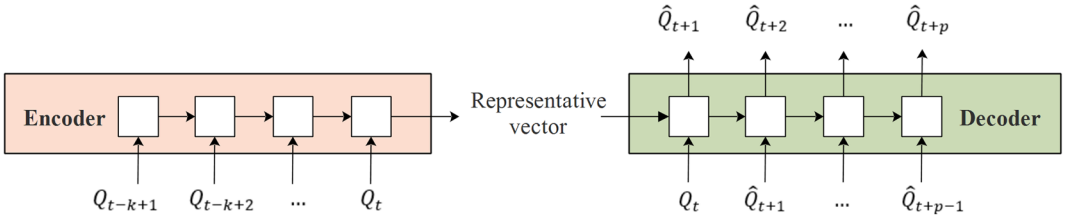


Fig. 1. Sequence to sequence model.

typically comprises two parts, an encoder and a decoder (Fig. 1). The encoder processes each item in the input sequence and captures useful features in that sequence in a representative vector. The representative vector, which is now a representation of the input sequence, is then fed to the decoder. The decoder then starts to produce the output sequence. The encoder and decoder are normally recurrent neural networks, although they can also be other types of neural networks.

In this way, the encoder has to fully capture the essence of the entire input sequence in a single hidden state. This is a very difficult task to achieve in practice, and this problem gets worse when the input sequence is longer. To deal with this, the attention mechanism was proposed on top of the sequence to sequence model (Bahdanau et al., 2014; Luong et al., 2015). The reasoning behind this is that when doing a prediction task, there are particular elements of the inputs that NNs need to pay more attention than the others. This is similar to when a human manually translates a long sentence, he/she would focus more on a specific word or phrase to translate at any given point.

Without attention, only the final state from the encoder is available to the decoder. Due to the sequential processing of the input sequence in the RNN encoder, this final state tends to keep more information about the most recent inputs than the older ones. In traffic state prediction, this might not be an issue when predicting the state of the next five minutes, as no considerable change would normally be expected in such a short period and the most recent state would then be very important. However, for longer prediction horizons like predicting the traffic state in the next one hour, the influences of the older states could be underestimated. Therefore, employing the attention mechanism over the sequence of previous states in predicting traffic flow could be a promising approach (see Fig. 2). A sequence to sequence model with GRUs used as the encoder and decoder applying such attention is thus proposed in this research and will be explained in more details in Section 3.5 below.

Using the attention mechanism, for each predicted step, a context vector is obtained based on the hidden states produced at every time step of the encoder as follows:

$$c_i = \sum_{j=1}^{k} a_{ij} s_j \tag{1}$$

$$a_{ij} = \frac{f(h_i, s_j)}{\sum_{j=1}^{k} f(h_i, s_j)} \tag{2}$$

where k is the length of the input sequence (which is also the number of hidden states of the encoder), c_i is the context vector for predicted time i, a_{ij} is the attention weight specifying how much hidden state j from the encoder should be attended when making prediction at time i, f is an attention function calculates an attention score between the hidden state at time i of the decoder h_i and hidden state j of the encoder s_j.

The attention mechanism could be generalized as a paradigm comprising of three parts including query, key and value. Query determines which pairs of key-value to focus on. Keys are used to calculated the attention scores and values are used to compute the context vector. In this case, h_i is query, s_j is both key and value.

Many functions have been proposed to be used as the attention function such as vector inner-product or dot product, general and concatenation (Luong et al., 2015). In our proposed model, dot function is chosen to be the attention function due to its simplicity and it is often as effective as other methods. The attention function f then becomes:

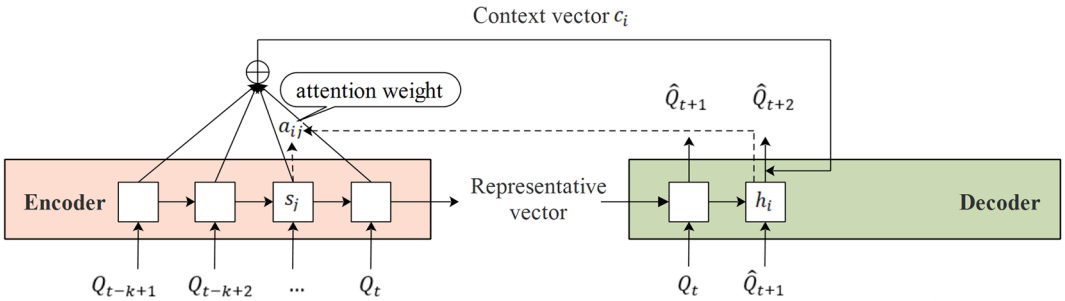


Fig. 2. Sequence to sequence model with attention.

$$f(h_i, s_j) = h_i^T s_j \tag{3}$$

The prediction is then made based on the context vector and the state of the decoder through a simple hidden layer concatenating the values (as in Luong et al., 2015) and an output layer as follows:

$$h_i^{'} = \tanh(W_a[c_i; h_i]) \tag{4}$$

$$\hat{y}_i = W_o h_i^{'} \tag{5}$$

where $h_i^{'}$ is the attentional hidden state, W_a and W_o are weight matrices, $\hat{y}_i$ is the predicted value at time i.

Note that the above calculations are made separately for each road link in the network, which means no spatial correlations have been explored. The spatial correlations will be exploited using spatial attentional matrices and convolutional GRUs which are introduced in Sections 3.3 and 3.4 below.

In the scope of this study, we utilize only the most commonly used input data for short-term traffic flow predictors, which is the data in previous steps. Other factors, such as daily periodicity (the same time of previous days) and weekly periodicity (the same day of previous weeks), could be considered in future work.

3.3. Spatial attention

The dependencies in traffic states among nearby locations have been utilized in many studies and are also explored in this work due to their significance in the evolution of traffic states. In our proposed model, these spatial dependencies are exploited by using the spatial attentional matrices. To handle the dynamics in spatial correlations, the spatial attentional matrices are time-dependent. The spatial attentional matrix $I^t \in \mathbb{R}^{N\times N}$ at time step t is defined as a two-dimensional matrix that meets the following conditions:

- All attention weights are from zero to one:

$$0 \le I_{i,j}^t \le 1, \forall\, v_i, v_j \in V \tag{6}$$

- $I_{i,j}^t$ represents the attention the node v_j requires from v_i at time t. The sum of the attention weights of one node to all nodes is one:

$$\sum_{j=1}^{N} I_{i,j}^t = 1, \forall\, v_i, v_j \in V \tag{7}$$

For each time step t, the attention weight measuring how much focus one node should have on another node's state to predict its future traffic flow is calculated. The calculation is based on the traffic state at that time, which is the DS^t value in this study. The DS values are chosen to extract spatial attention since they better track the network traffic states than other quantity such as traffic volumes. Additionally, compared to use only traffic volume, using a second traffic variable could provide the predictor with more information about the traffic conditions. The relationship between these variables, which is one of the fundamental diagram relationships (an important feature of traffic), could also help. The process of generating the spatial attentional matrix is illustrated in Fig. 3. The input DS values with c_{in} input channels of each ordered pair of nodes (road segments) are convoluted to yield spatial correlation features with c_{out} output channels. By multiplying with the U-th order adjacency matrix A^U, only the features between neighboring nodes are kept for further processing. The output channels of the resulting matrix are then combined to produce the spatial attentional matrix.

Firstly, given a directed graph G, we construct an adjacency matrix $A \in \mathbb{R}^{N\times N}$, $A_{i,j} = 1$ if there is an edge from node v_j to node v_j. In a traffic network, there should always exists a path between every pair of road links. Then, the distance function $d(v_i, v_j)$ in the graph G can be defined as the minimum number of edges needed to traverse from node v_i to node v_j. The U-th order adjacency matrix $A^U \in \mathbb{R}^{N\times N}$ is then defined as:

$$A_{i,j}^U = \begin{cases} 1, & d(v_i, v_j) \le U \vee d(v_j, v_i) \le U \\ 0, & \text{otherwise} \end{cases}, \quad \forall\, v_i, v_j \in V \tag{8}$$

As the$D$-product of A can generate a matrix A^D in which $A_{i,j}^D > 0$ if v_j can be traversed from v_i using exact D edges (Newman, 2010), A^U can be calculated based on the definition (8) as follows:

$$\hat{A}^U = \varphi(A^1 + A^2 + \cdots + A^U) = \varphi\left(A + \prod_{i=1}^{2} A + \cdots + \prod_{i=1}^{U} A\right) \tag{9}$$

$$A^U = \max\left(\hat{A}^U, \hat{A}^{U^T}\right) \tag{10}$$

where φ is a function that clips elements greater than 1 to 1, so all elements $\in\{0, 1\}$. This function is simply a rounding down function which takes a value x and returns x if x is less than or equal to 1, otherwise it returns 1. $\hat{A}^{U^T}$ is the transpose of $\hat{A}^U$. Eq. (10) is used to convert the matrix resulting from (9) to a symmetric matrix, as both upstream and downstream links are of interest. Based on A^U, the neighborhood of a node $v_i \in V$ could be defined as the set $\{v_j \in V | A_{i,j}^U = 1\}$. In the proposed model, A^U is taken into account when

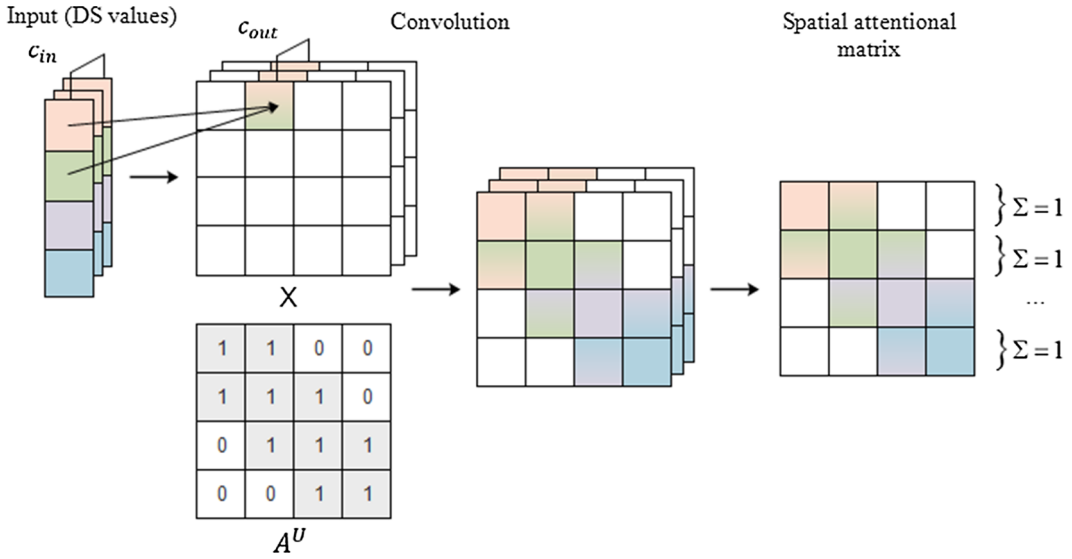


Fig. 3. Spatial attention extraction.

constructing spatial attentional matrices. More specifically, if $A^U_{i,j} = 1$, the attention v_j requires from v_i is calculated and used for further steps.

Then, for an ordered pair of nodes $(v_i, v_j) \in V$, the attention score v_j requires from v_i at time t is calculated as follows:

$$\widehat{C}^t_{i,j} = DS^t_i \cdot W_{c1} + DS^t_j \cdot W_{c2} \tag{11}$$

$$C^t_{i,j} = \sigma(\widehat{C}^t_{i,j} \circ A^U_{i,j}) \cdot W_{co} \tag{12}$$

where W_{c1}, $W_{c2} \in \mathbb{R}^{c_{in}, c_{out}}$ are filters with c_{in} input channels (number of DS values for each node at each time step used as inputs) and c_{out} output channels. As the influences between each pair of nodes are not supposed to be symmetric, these two different filters make it possible for the attention scores of (v_i, v_j) and (v_j, v_i) to be different. $W_{co} \in \mathbb{R}^{c_{out}, 1}$ is a weight matrix, the $\cdot$ operator is matrix multiplication, the $\circ$ operator is element-wise multiplication. σ is an activation function and Rectified Linear Unit (ReLU) function is chosen in this model as it is recommended for convolutional layers (Krizhevsky et al., 2012).

As the spatial attentional matrices are required to meet the conditions (6) and (7), the softmax activation function is applied along each row i of the matrix resulting from the previous step ($C^t \in \mathbb{R}^{N \times N}$) to produce the spatial attentional matrix (at time t) I^t.

$$I^t_i = \text{softmax}(C^t_i) \tag{13}$$

3.4. Convolutional GRU

In this study, convolutional GRUs are used to capture both the temporal and spatial dependencies of traffic data. The structure of a convolutional GRU remains the same as a standard GRU with an update gate and a reset gate. The update gate decides how much of the previous state is retained. The reset gate determines how the new input is fused with the previous state. In a convolutional GRU, to produce the next state for a node in V (a road segment), the features of the neighboring nodes are fused with its own features through the convolutional operation. The convolutions are based on a weighted adjacency matrix representing the spatial correlations between nearby nodes, such as a spatial attentional matrix.

The convolutional operation using matrix $C \in \mathbb{R}^{N \times N}$ is then defined as:

$$W \odot^C B = [B; C \cdot B] \cdot W \tag{14}$$

where W is a weight matrix, B is the input matrix and C is a weighted adjacency matrix.

The state of the convolutional GRU is then calculated as follows. The pure matrix multiplication in a standard GRU is replaced with the convolutional operation.

$$z_t = \sigma_g(W_z \odot^C X^t + U_z \odot^C h_{t-1} + b_z) \tag{15}$$

$$r_t = \sigma_g(W_r \odot^C X^t + U_r \odot^C h_{t-1} + b_r) \tag{16}$$

$$h_t = \sigma_h(W_h \odot^C X^t + U_h \odot^C (r_t \circ h_{t-1}) + b_h) \tag{17}$$

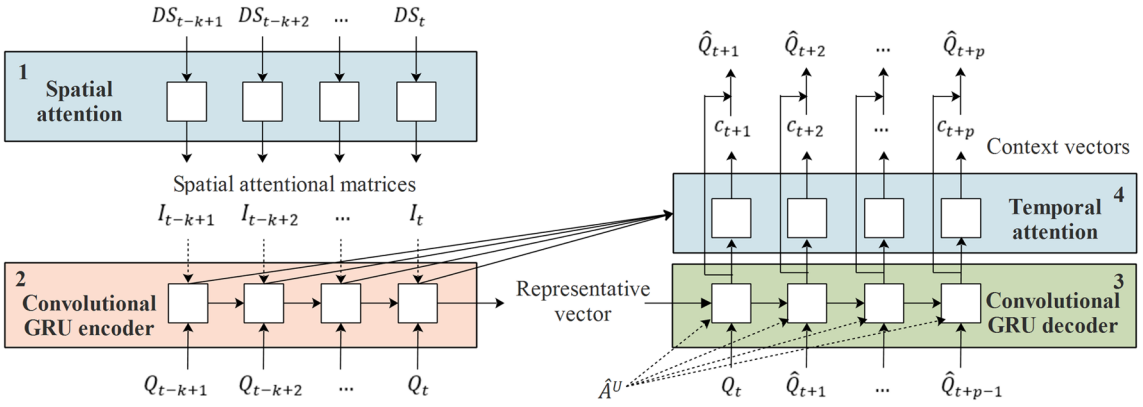


Fig. 4. Spatial-temporal attention neural network (STANN).

$$h_t = (1 - z_t)^{\circ} h_{t-1} + z_t^{\circ} h_t \tag{18}$$

where z_t is the update gate, r_t is the reset gate, h_t is the current memory content, h_t is the output of the convolutional GRU, σ_g is a sigmoid function and σ_h is a hyperbolic tangent function as in a conventional GRU, W_z, W_r and W_h are weight matrices connecting input to the two gates and the current memory content, U_z, U_r and U_h are weight matrices connecting the previous output state to the gates and the current memory content, b_z, $b_r$andb_h are bias vectors, the $\odot^C$ operator is the convolutional operation using matrix C.

3.5. Spatial-temporal attention based neural network model (STANN)

The proposed prediction model STANN is a sequence to sequence model with both temporal and spatial attention (Fig. 4).

In the proposed model, although both the encoder and decoder use convolutional operations, the convolutions are based on different matrices. The convolution in the encoder is based on the spatial attentional matrices as the traffic states in previous time steps are available to the encoder and could be utilized. However, the decoder has no information regarding the traffic states in the prediction horizon, so the weighted U-th order adjacency matrix is used for convolution. This matrix captures static spatial correlations as it is not time-dependent, unlike the spatial attentional matrix. It is defined as:

$$\widehat{A}^U = A^{U\,\circ} W_A \tag{19}$$

where A^U is the U-th order adjacency matrix and W_A is a weight matrix which is trained together with the model.

The work flow of the model could be described as follows.

1. The convolutional layer takes DS values at k previous time steps DS^{t-k+1}, DS^{t-k+2}, ⋯, DS^t as inputs. For each time step, a spatial attentional matrix is produced. Then there are k spatial attentional matrices constructed I^{t-k+1}, I^{t-k+2}, ⋯, I^t.
2. The encoder takes traffic volume values at k previous time steps Q^{t-k+1}, Q^{t-k+2}, ⋯, Q^t as inputs. The encoder, which is a convolutional GRU, processes through this input sequence. For each time step, the corresponding spatial attentional matrix constructed for that time step is used for the convolution in the GRU. The encoder then outputs k states for k previous time steps. The final state is used as the initial state for the decoder.
3. The decoder produces an output sequence one by one. The weighted U-th order adjacency matrix $\widehat{A}^U$ is used for the convolution at every time step in the decoder. For each output step, the decoder takes ground truth or predicted traffic volume in the previous step as input. For the testing stage, only ground truth values up to the current time t are available, so Q^t is used for the first output step and predicted values are used for the other steps. For the training stage, the choice of using ground truth values or predicted values is determined based on a probability.
4. For each output step, the attention weights of the k states generated by the encoder (keys) are calculated with regard to the output of the decoder for that step (query). The context vector is constructed based on the attention weights, which is then used together with the output state from the decoder to make the final prediction for that step. This process is repeated until the required prediction horizon is achieved.

3.6. The utilization of different data resolutions

For a node, each element of the input sequence fed to the encoder is just one volume value, which is in very low dimensionality. Although the convolutions in the GRU encoder using the spatial attentional matrices could help increase the dimensionality of the input by producing additional values taking from neighboring nodes, the inputs could still be further improved. One possible method is using the data of higher resolutions as inputs. To be more specific, if the model aims to predict x-min. traffic volumes, instead of using only x-min. traffic volumes as inputs to the encoder, z values of y-min. traffic volumes, where $y \times z = x$, are fed to the encoder. These values are scaled (multiplied by z) to match the range of x-min. traffic volumes. When extracting spatial attention, z values of

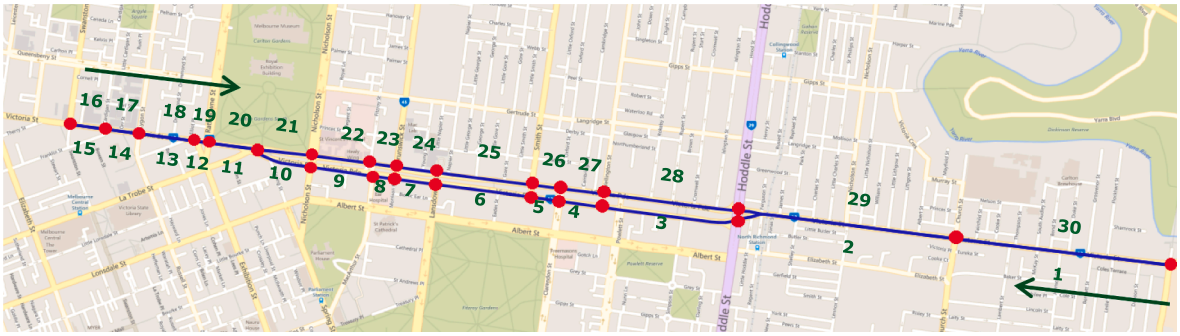


Fig. 5. The studied network. Note: The green arrows indicate the directions. The road segments are formed by the marked intersections and their indexes are shown in the figure. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

y-min. DS instead of one x-min. DS value are also used as inputs for the convolutional layer, which means the number of input channels c_{in} is z. In this way, the dimensionality of the inputs is increased by z times and all the data still belong to the road link itself, so the data would be relevant and useful for the prediction.

4. Experiments

4.1. Data description

For the experiments, we use the SCATS traffic volume dataset provided by VicRoads. Our studied network is an urban corridor of 30 road segments with 24 intersections along Victoria Street (Melbourne) including both directions (Fig. 5). Victoria Street is one of the busiest streets in Melbourne so the prediction of traffic flows for this street is of high interest. The dataset contains traffic volume and DS of every road segment per minute. One-year data of the year 2016 are used in the experiments; public holidays and the days with missing data are excluded. The data are aggregated into 5-min. intervals and normalized to the range [0, 1] using the min-max normalizing technique. In this study, we focus on evaluating the models' performance in the morning peak hours (6 am–10 am) on weekdays. The morning peak is one of the periods that has attracted a lot of attention due to its high traffic volume and level of congestion (Ishak et al., 2003; Chan and Dillon, 2012). Besides, this period includes both the traffic transition states before and after the peak traffic state, which makes the period challenging for predictors and interesting to study. The first 60% of the data is used for training, the next 20% is used for validation and the last 20% is used for testing.

4.2. Experimental settings

4.2.1. Baselines

We compare the performance of the proposed model (STANN) with SVR (Support Vector Regression), RFR (Random Forest Regression), Feedforward Neural Network (FFNN), GRU (Gated Recurrent Unit), LSTM (Long-short Term Memory), Seq2Seq (a basic sequence to sequence model), SAE (Lv et al., 2015), DCRNN (Li et al., 2018) and DNN-BTF (Wu et al., 2018).

- SVR: Apart from NN-based models, SVR is also a popular machine learning approach for traffic flow prediction, so a basic SVR model is selected as one of our baselines.
- RFR: A basic RFR model, a decision tree based model which has been shown to be effective for the task, is also employed as a baseline.
- FFNN/GRU/LSTM: These models have one hidden layer with 100 hidden units. ReLU is used as the activation function for the hidden layer in FFNN, and GRU/LSTM utilizes the activation functions as in the original structures.
- Seq2Seq: The encoder and decoder in this sequence to sequence model are GRUs. In this model, the calculations are made separately for each road segment, which means no spatial correlations are explored. The GRUs have one hidden layer with 32 hidden units, which is the optimal configuration obtained for this model. As our proposed model makes use of this model, this is the main baseline in the experiments.
- SAE: This model has four stacked autoencoder layers with 300 hidden units in each layer as suggested by the original work.
- DCRNN: This model is based on Seq2Seq's architecture. It employs diffusion matrices to convolute data in the GRU encoder and decoder. The source code of DCRNN is available online so we simply use this to test the model. The model structure is as described in the original work. The GRUs have two hidden layers with 64 hidden units.
- DNN-BTF: In Wu et al. (2018), DNN-BTF employs data from previous days and previous weeks in addition to data of previous time steps. However, for fair comparisons, only the data of previous time steps are used for this model, as with other models in our experiments. Also, the use of speed and flow data in the original work are replaced with DS and volume. Apart from these modifications, DNN-BTF is implemented strictly based on the description.

Except for the main baseline (Seq2Seq), the structures of other baselines are empirically chosen or constructed based on the original works.

4.2.2. Model structures

In our experiments, we develop two models STANN and STANN*. STANN* is similar to STANN except different data resolutions are used as inputs, as described in Section 3.6. STANN utilize only 5-min. interval data while STANN* utilize 5 values of 1-min. interval data (non-aggregated data) as inputs to the encoder and to extract spatial attention ($x = 5, y = 1, z = 5$). The hyperparameters of the models are obtained by performing grid search runs. The order of neighborhood U is chosen as 6, the number of hidden units in the convolutional GRUs and the number of output channels c_{out} in the convolutional layer are 32. In the training stage, the ratio of the predicted values used as inputs for the decoder, instead of ground truth values, is selected as 0.75 through similar runs.

The implementations of the neural network models are done using the Tensorflow and the SVR and the RFR are implemented using Scikit-Learn. The prediction horizon is up to 1 h (12 steps) and the number of looked-back steps k is 36 (3 h before the prediction time) for all the models. All the neural network models are trained using the Adam optimizer (Kingma and Ba, 2014) with a batch size of 32. The Mean Absolute Error (MAE) is used as the objective function. The initial learning rate is 0.01 and the early stopping technique is used to avoid overfitting.

4.2.3. Evaluation metrics

To evaluate the models' performance, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Weighted Mean Absolute Percentage Error (WMAPE), which are popular measures for prediction models, are used.

$$MAE = \frac{\sum_{i=1}^{n}\sum_{j=1}^{N}\left|y_{i,j}^{'} - y_{i,j}\right|}{n \times N} \tag{20}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{N}\left(y_{i,j}^{'} - y_{i,j}\right)^{2}}{n \times N}} \tag{21}$$

$$WMAPE = 100 \times \frac{\sum_{i=1}^{n}\sum_{j=1}^{N}\left|y_{i,j}^{'} - y_{i,j}\right|}{\sum_{i=1}^{n}\sum_{j=1}^{N}\left|y_{i,j}\right|} \tag{22}$$

$y_{i,j}^{'}$: the predicted output of the j^{th} road segment in the i^{th} sample
$y_{i,j}$: the target output of the j^{th} road segment in the i^{th} sample
n: the number of samples
N: the number of road segments

4.3. Results

4.3.1. Prediction accuracy

The comparison of the prediction errors in different prediction steps can be seen in Table 1. With simpler architectures compared to the other methods (deep learning models), it is not surprising to see that SVR, RFR and FFNN have the highest errors. Among these models, RFR has proven to be the most promising approach for this task as it demonstrates its superiority to SVR and has a slightly better performance than FFNN. As SAE fuses all information of the links and time steps together, which would prevent the model from capturing useful and complex spatial-temporal features, the only NN-based model it outperforms is FFNN. Exploring the temporal dependencies, GRU and LSTM models are shown to be more accurate than the above models. The performance of these models are comparable to each other, which is consistent with the literature (Fu et al., 2016). Making use of an attention model in which both spatial and temporal correlations are exploited, DNN-BTF achieves an encouraging accuracy but it does not clearly outperform GRU and LSTM. The construction of the attention matrix in this model simply uses fully-connected layers taking all road links into consideration in the first place, which could make it harder and less efficient for the model to identify the most relevant links. The sequence to sequence structure is shown to be a good choice for the task as all the models using this structure perform well in the experiments. Though the basic sequence to sequence model does not utilize any information from other road links, it is still comparable with DNN-BTF. Among the baselines, DCRNN obtains a very promising accuracy in the experiments. Although STANN and DCRNN shares a similar structure, the use of spatial and temporal attention, instead of the static diffusion matrices constructed based solely on the network topology, helps improve the model's performance. As can be seen from the results, our proposed model, STANN, demonstrates its superiority over the baselines in all the prediction steps and all the evaluation metrics.

Student's t-tests were conducted between the proposed model and the baselines to see whether the differences between the models are statistically significant. With p-values $< 1e-4$ when comparing with all the baselines, STANN can be seen to be statistically significantly better than the baselines with a significance level of 99.99%.

Additionally, the use of different data resolutions as inputs in STANN* helps improve the overall performance of the model. This improvement could be explained by the additional helpful information provided to the model through this technique.

Table 1
Prediction errors of STANN, STANN* and all the baselines with 95% confidence intervals. The results of our proposed models are marked bold.

Prediction step	Step 1 (5 min)			Step 3 (15 min)			Step 6 (30 min)			Step 12 (60 min)			Average		
	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE
SVR	21.2	8.7	11.1	21.3	8.9	11.4	21.4	9.1	11.7	22.1	9.6	12.4	21.5	9.1	11.7
RFR	19.2	7.8	10.6	20.1	8.4	11.2	21.0	8.9	11.8	22.2	9.6	12.7	20.6	8.7	11.6
FFNN	20 ± 0.1	8.2	11.3 ± 0.1	20.1 ± 0.1	8.3	11.4	20.7 ± 0.1	8.8	11.8 ± 0.1	22.9 ± 0.4	9.9 ± 0.2	13.2 ± 0.2	20.9 ± 0.1	8.8 ± 0.1	11.9 ± 0.1
SAE	19.4 ± 0.1	7.9	10.9	19.6 ± 0.1	8.1	11.1	20.4 ± 0.1	8.7	11.7	21.8 ± 0.1	9.4	12.6 ± 0.1	20.3 ± 0.1	8.5	11.6
GRU	18.3	7.5	10.3	19.3	8.0	11	20.3	8.6	11.6	21.5 ± 0.1	9.3	12.7 ± 0.1	19.9	8.4	11.4
LSTM	18.4 ± 0.1	7.5	10.3	19.4	8.0	10.9	20.5 ± 0.1	8.7	11.6	21.7 ± 0.1	9.4	12.5	20 ± 0.1	8.4	11.3
DNN-BTF	18.9 ± 0.1	7.7 ± 0.1	10.6 ± 0.1	19.2 ± 0.1	8.0	10.9 ± 0.1	20.2 ± 0.1	8.5	11.5 ± 0.1	21.1 ± 0.1	9.1	12.3 ± 0.1	19.8 ± 0.1	8.3	11.3
Seq2Seq	17.5 ± 0.1	7.3	9.8	18.9 ± 0.1	7.8	10.5 ± 0.1	19.3 ± 0.1	8.3	11.2 ± 0.1	22.3 ± 0.1	9.7 ± 0.1	13.4 ± 0.1	19.5 ± 0.1	8.2	11.3 ± 0.1
DCRNN	16.6 ± 0.2	6.7 ± 0.1	9.3 ± 0.1	17.5 ± 0.1	7.2	9.8	18.5 ± 0.1	7.9	10.6	20 ± 0.1	8.6	11.4	18.2 ± 0.1	7.6	10.3
STANN	**15.8 ± 0.1**	**6.5**	**8.8**	**16.3**	**6.8**	**9.2**	**17.2 ± 0.1**	**7.3**	**9.8**	**18.8 ± 0.1**	**8.1**	**10.8**	**17.0 ± 0.1**	**7.2**	**9.7**
STANN*	**14.2**	**5.8**	**8.1**	**15.6**	**6.5**	**8.9**	**16.8**	**7.1**	**9.7**	**18.6 ± 0.1**	**8.0**	**10.7**	**16.3**	**6.9**	**9.4**

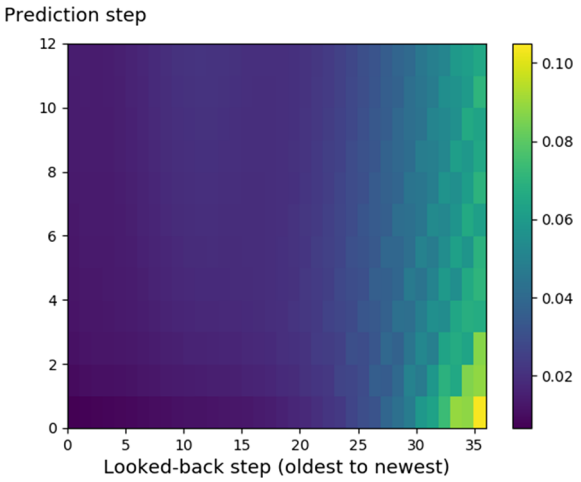
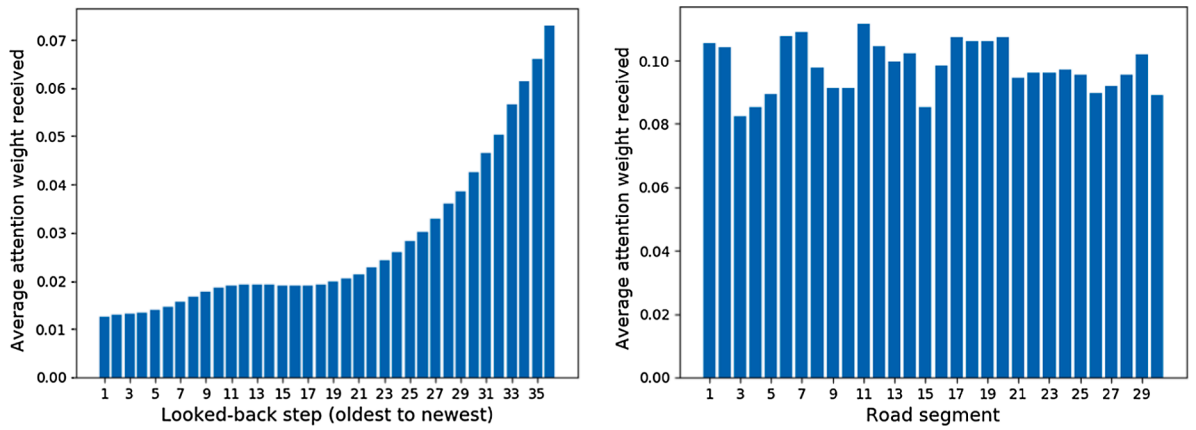


Fig. 6. Average temporal attention weights.

4.3.2. Temporal attention

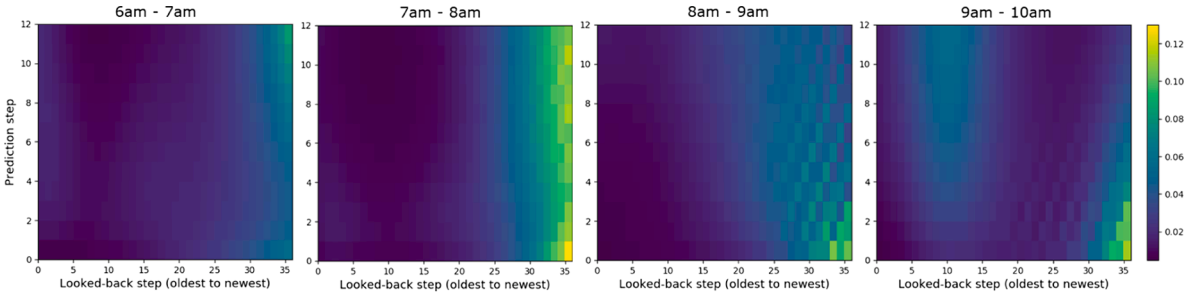
A visualization of the average temporal attention weights can be seen in Fig. 6. The average attention weights of each looked-back step for each prediction step are calculated over the whole test set. As can be seen from the heat map, the most recent steps are the most relevant to the prediction of all steps. However, the attention received by the most recent steps are higher in the prediction of the shorter horizons. On the other hand, the significance of the further previous steps is slightly increased as the prediction horizon is longer. These observations are consistent with our very first hypothesis when suggesting the use of the attention mechanism over the looked-back steps to avoid underestimating the steps other than the closest previous steps. Additionally, the importance of each looked-back step can be measured by the average attention received (Fig. 7(a)). The most recent steps are the most important ones and the importance of the steps generally decrease as the steps get further from the current step. However, it is noticeable that the decrease is not linear and there are some consecutive steps (11–18) where the attention weights seem to be nearly the same.

Furthermore, it can be observed that the temporal correlations exploited in different hours are not the same (see Fig. 8(a)). This shows the capability of the model in capturing dynamic temporal dependencies. As the traffic in a few hours before does not help much with the predictions in 6 am–7 am, the temporal relationship in this period is not as clear as other periods. For the period of 7 am–8 am, it can be seen that the predictions are based mostly on the traffic state in the previous hour (i.e. 6 am–7 am). It could be because the traffic state before that (4 am–6 am) usually does not provide very applicable information about the traffic in the morning peak. The predictions for 8 am–10 am utilize the traffic in the older steps more. While the temporal correlations in 8 am–9 am are as expected (similar to the average attention), the relationships can become very complicated as with 9 am–10 am, in which the attention weights of step 7 to step 18 are surprisingly higher than the closer steps. This makes studying the temporal relationships exploited by deep learning models in traffic prediction interesting.

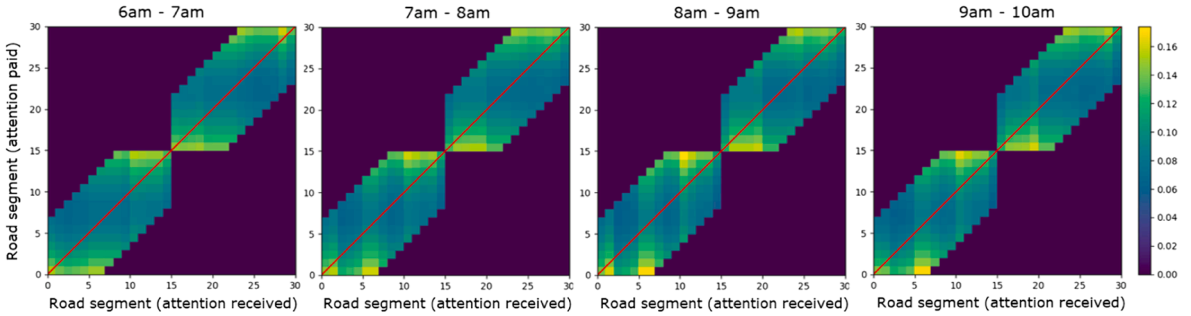


(a) Temporal attention by looked-back steps (b) Spatial attention by road segments

Fig. 7. Average attention received by different looked-back steps and different road segments.



(a) Temporal attention over time



(b) Spatial attention over time

Fig. 8. Spatial-temporal attention in different hours.

4.3.3. Spatial attention

An example of the visualization of the spatial attention weights can be found in Fig. 9. The heat maps show the average attention between each pair of neighboring road segments. It can be seen that the attention weights of the lower order of neighboring road segments are higher in many cases such as for road segments 15–20. However, this is not always true (such as for road segments 21–25), which means the spatial dependencies between road links would not be simply based on the distance between road links. Road link 1, 15, 16 and 30 are at the boundary of the chosen corridor, so the number of neighboring links is smaller. This leads to the higher attention paid from these links, as the sum of attention weights of one link is one.

The visualization of attention weights can help us not only to see how much attention is paid and received between each pair of road segments, but also to identify the most important road segments. The importance of a road segment in this context could be understood as how much influence the road segment has on other road segments. More specifically, one road segment could be interpreted to be more important than another one if it gains higher attention when predictions are made for other road segments. For

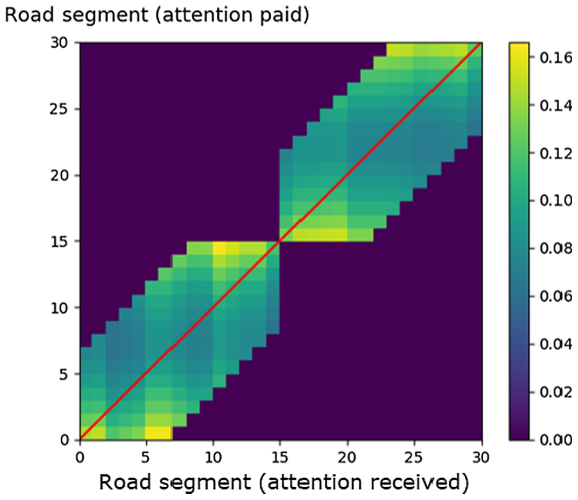


Fig. 9. Average spatial attention weights.

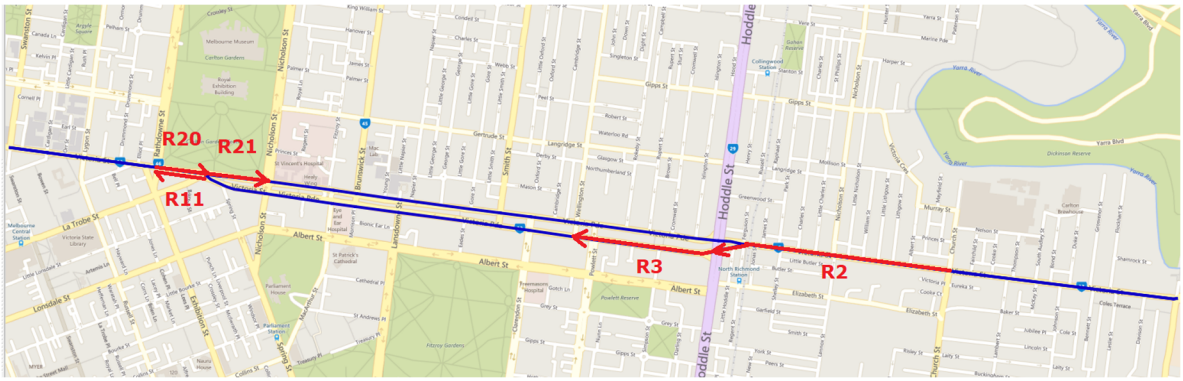


Fig. 10. The studied network with marked road segments (R = Road segment).

this purpose, the average spatial attention received by each road segment is visualized in Fig. 7(b). As can be seen in both Figs. 7(b) and 9, there are a few road segments requires a lot more attention from its nearby links than the others. For example, road segments 2 and 20 are clearly shown to be more important than road segments 3 and 21. The importance of these two road segments could be explained by the network topology. These road segments are connected with big intersections, where a lot of vehicles from other road links can join Victoria Street at that point and propagate along the street. Similarly, road segment 11, which is connected with another big intersection, also obtained very high attention from other links. The road segments 2, 3, 20, 21 and 11 are marked on the map in Fig. 10.

Additionally, the dynamics in spatial dependencies captured by the proposed model are demonstrated through the difference between the average attention weights in different hours (Fig. 8(b)). Compared to the other hours, the attention received by the road segments appears to be less varied in the period between 6am and 7am, specifically as observed in the road segments 1–10. This might indicate that in this period the spatial correlations are less relevant. The similarity in the spatial relationships during the morning hours from 7 am to 10 am could also be revealed by the similarity in the patterns of spatial attention.

The above analyses show the effectiveness of the attention mechanism applied in STANN in improving the neural network model's interpretability and revealing spatial-temporal traffic correlations, which could be helpful for traffic management.

5. Conclusions

In this research, a deep learning based traffic flow prediction model utilizing spatial and temporal attention, called STANN, is proposed. Inspired by the idea of attention-based models, the spatial and temporal attentions are proposed to be used for traffic flow prediction in this study. Experiments are conducted on a real-world traffic dataset and the experiment results demonstrate the encouraging performance of the proposed model. It indicates the potential of applying the attention mechanism to exploit the spatial-temporal dependencies in traffic. The experiments also show the effectiveness of utilizing higher-resolution data to improve the proposed model's overall accuracy when predicting lower-resolution data. Additionally, the proposed model has been shown to be capable of giving insights about the exploited spatial-temporal traffic correlations.

For future work, it would be interesting to evaluate the effectiveness of the proposed model on larger and more complex traffic networks. With the ability to explore dynamic spatial-temporal correlations through the attention mechanism, the next step could be investigating the adaptability of this type of model for a wide range of evolving traffic conditions (e.g. changes in traffic during normal peak hours, on public holidays and weekends, and during planned and unplanned disruptions). Additionally, further time dependency of the spatial attentional matrix (e.g. utilizing multiple DS values through a moving window approach to calculate the matrix) could be considered to improve the results. The improvement resulting from the use of different data resolutions in this study also motivates further applications of this technique.

Appendix A

In this study, although the period of the morning peak hours on weekdays was adopted in the experiments, our proposed model should be applicable for other periods. To show the proposed model's performance in another challenging period, more experiments were conducted using the data in the afternoon peak hours (3 pm–7 pm) on weekdays to train and test the model. Compared to Seq2Seq, the main baseline on which out proposed model is built, STANN and STANN* still demonstrate their superiority (see the results in Table 2). This confirms the effectiveness of the proposed spatial-temporal attention mechanisms. In addition, consistent with the morning peak, the utilization of different data resolutions can also help improve the proposed model's performance in the afternoon peak. Compared to the morning peak, the afternoon peak is also shown to be very challenging (Table 3). The errors in the afternoon peak are generally higher than in the morning peak, However, the difference in the average WMAPEs, which takes into account the percentage errors (relative errors), is not that significant.

Table 2
Prediction errors of Seq2Seq, STANN and STANN* in the afternoon peak with 95% confidence intervals. The average errors are marked bold.

	Seq2Seq				
	Step 1 (5 m)	**Step 3 (15 m)**	**Step 6 (30 m)**	**Step 12 (60 m)**	**Avg.**
WMAPE	17.2	18.2	18.2	21.1	**18.7**
MAE	7.3	8.3	8.4	9.3	**8.5**
RMSE	10.2	10.7	10.7	12.0	**10.9**
	STANN				
	Step 1 (5 m)	**Step 3 (15 m)**	**Step 6 (30 m)**	**Step 12 (60 m)**	**Avg.**
WMAPE	15.1 ± 0.1	16.3 ± 0.1	17.7 ± 0.1	20.2 ± 0.1	**17.3 ± 0.1**
MAE	6.9	7.5	8.1	9	**7.9**
RMSE	9.2	9.8	10.4 ± 0.1	11.6 ± 0.1	**10.2**
	STANN*				
	Step 1 (5 m)	**Step 3 (15 m)**	**Step 6 (30 m)**	**Step 12 (60 m)**	**Avg.**
WMAPE	13.4	15.2	16.8	19.4 ± 0.1	**16.2**
MAE	6.1	7.0	7.6	8.6	**7.3**
RMSE	8.4	9.3	10.1	11.2 ± 0.1	**9.7**

Table 3
The comparisons of the proposed models' average prediction errors in morning peak and afternoon peak (with 95% confidence intervals).

	Morning peak			Afternoon peak		
	WMAPE	MAE	RMSE	WMAPE	MAE	RMSE
STANN	17.0 ± 0.1	7.2	9.7	17.3 ± 0.1	7.9	10.2
STANN*	16.3	6.9	9.4	16.2	7.3	9.7

References

Bachrach, Y., Zukov-Gregoric, A., Coope, S., Tovell, E., Maksak, B., Rodriguez, J., McMurtie, C., 2017. An attention mechanism for answer selection using a combined global and local view. arXiv preprint arXiv:1707.01378.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Barbieri, F., Anke, L.E., Camacho-Collados, J., Schockaert, S., Saggion, H., 2018. Interpretable emoji prediction via label-wise attention lstms. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4766–4771.

Brown, A., Tuor, A., Hutchinson, B., Nichols, N., 2018. Recurrent neural network attention mechanisms for interpretable system log anomaly detection. arXiv preprint arXiv:1803.04967.

Chan, K.Y., Dillon, T.S., 2012. On-road sensor configuration design for traffic flow prediction using fuzzy neural networks and taguchi method. IEEE Trans. Instrum. Meas. 62 (1), 50–59.

Chen, C., Hou, J., Shi, X., Yang, H., Birchler, J.A., Cheng, J., 2019. Interpretable attention model in transcription factor binding site prediction with deep neural networks. bioRxiv 648691.

Chen, H., Grant-Muller, S., Mussone, L., Montgomery, F., 2001. A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. Neural Comput. Appl. 10 (3), 277–286. https://doi.org/10.1007/s521-001-8054-3.

Cheng, X., Zhang, R., Zhou, J., Xu, W., 2018. Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.

Chiu, C.C., Sainath, T.N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Jaitly, N., 2018. State-of-the-art speech recognition with sequence-to-sequence models. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4774–4778.

Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1724–1734.

Cui, Z., Henrickson, K., Ke, R., Wang, Y., 2018. High-order graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting. arXiv preprint arXiv:1802.07007.

Dai, X., Fu, R., Zhao, E., Zhang, Z., Lin, Y., Wang, F.Y., Li, L., 2019. DeepTrend 2.0: a light-weighted multi-scale traffic prediction model using detrending. Transp. Res. Part C: Emerg. Technol. 103, 142–157.

Deng, S., Jia, S., Chen, J., 2019. Exploring spatial–temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data. Appl. Soft Comput. 78, 712–721.

Do, L.N., Taherifar, N., Vu, H.L., 2019. Survey of neural network-based models for short-term traffic state prediction. Wiley Interdisc. Rev.: Data Min. Knowl. Discovery 9 (1), e1285.

Ermagun, A., Levinson, D., 2018. Spatiotemporal traffic forecasting: review and proposed directions. Transp. Rev. 38 (6), 786–814.

Ermagun, A., Levinson, D., 2019. Spatiotemporal short-term traffic forecasting using the network weight matrix and systematic detrending. Transp. Res. Part C: Emerg. Technol. 104, 38–52.

Fouladgar, M., Parchami, M., Elmasri, R., Ghaderi, A., 2017. Scalable deep traffic flow neural networks for urban traffic congestion prediction. In: 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 2251–2258. https://doi.org/10.1109/ijcnn.2017.7966128.
Fu, R., Zhang, Z., Li, L., 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In: Youth Academic Annual Conference of Chinese Association of Automation (YAC). IEEE, pp. 324–328.
Guan, D., Huang, L., Qu, Q., 2018. A predicting method of urban traffic network volume based on STARIMA model. In: CICTP 2017: Transportation Reform and Change—Equity, Inclusiveness, Sharing, and Innovation. American Society of Civil Engineers, Reston, VA, pp. 3600–3606.
He, S., Hu, C., Song, G.J., Xie, K.Q., Sun, Y.Z., 2008. Real-time short-term traffic flow forecasting based on process neural network. In: International Symposium on Neural Networks. Springer, Berlin, Heidelberg, pp. 560–569. https://doi.org/10.1109/wcica.2010.5554911.
Huang, W., Song, G., Hong, H., Xie, K., 2014. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. 15 (5), 2191–2201. https://doi.org/10.1109/tits.2014.2311123.
Ishak, S., Kotha, P., Alecsandru, C., 2003. Optimization of dynamic neural network performance for short-term traffic prediction. Transp. Res. Rec.: J. Transp. Res. Board 1836, 45–56.
Jia, Y., Wu, J., Ben-Akiva, M., Seshadri, R., Du, Y., 2017. Rainfall-integrated traffic speed prediction using deep learning method. IET Intel. Transport Syst. 11 (9), 531–536. https://doi.org/10.1049/iet-its.2016.0257.
Jia, Y., Wu, J., Du, Y., 2016. Traffic speed prediction using deep learning method. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1217–1222. https://doi.org/10.1109/itsc.2016.7795712.
Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: differences, similarities and some insights. Transp. Res. Part C: Emerg. Technol. 19 (3), 387–399.
Kingma, D.P., Ba, J.L., 2014. Adam: a method for stochastic optimization. Proc. 3rd Int. Conf. Learn. Representations.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.
Laña, I., Lobo, J.L., Capecci, E., Del Ser, J., Kasabov, N., 2019. Adaptive long-term traffic state estimation with evolving spiking neural networks. Transp. Res. Part C: Emerg. Technol. 101, 126–144.
Leng, Z., Gao, J., Qin, Y., Liu, X., Yin, J., 2013. Short-term forecasting model of traffic flow based on GRNN. In: 25th Chinese Control and Decision Conference (CCDC). IEEE, pp. 3816–3820. https://doi.org/10.1109/ccdc.2013.6561614.
Li, Y., Yu, R., Shahabi, C., Liu, Y., 2018. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. International Conference on Learning Representations (ICLR) 2018.
Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transp. Res. Part C: Emerg. Technol. 97, 258–276.
Lingras, P., Mountford, P., 2001. Time delay neural networks designed using genetic algorithms for short term inter-city traffic forecasting. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, Berlin, Heidelberg, pp. 290–299. https://doi.org/10.1007/3-540-45517-5_33.
Luong, M.T., Pham, H., Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16 (2), 865–873.
Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17 (4), 818. https://doi.org/10.3390/s17040818.
Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015a. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C: Emerg. Technol. 54, 187–197. https://doi.org/10.1016/j.trc.2015.03.014.
Ma, X., Yu, H., Wang, Y., Wang, Y., 2015b. Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS ONE 10 (3), e0119044. https://doi.org/10.1371/journal.pone.0119044.
Newman, M., 2010. Networks: An Introduction. Oxford University Press.
Pan, T.L., Sumalee, A., Zhong, R.X., Indra-Payoong, N., 2013. Short-term traffic state prediction based on temporal–spatial correlation. IEEE Trans. Intell. Transp. Syst. 14 (3), 1242–1254.
Rush, A.M., Chopra, S., Weston, J., 2015. A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transp. Res. Part C: Emerg. Technol. 79, 1–17. https://doi.org/10.1016/j.trc.2017.02.024.
Smith, B.L., Demetsky, M.J., 1994. Short-term traffic flow prediction: Neural network approach. Transp. Res. Rec. 1453, 98–104.
Sun, S., Huang, R., Gao, Y., 2012. Network-scale traffic modeling and forecasting with graphical lasso and neural networks. J. Transp. Eng. 138 (11), 1358–1367. https://doi.org/10.1061/(asce)te.1943-5436.0000435.
Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112.
Van Der Voort, M., Dougherty, M., Watson, S., 1996. Combining Kohonen maps with ARIMA time series models to forecast traffic flow. Transp. Res. Part C: Emerg. Technol. 4 (5), 307–318.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008.
Vlahogianni, E.I., 2015. Optimization of traffic forecasting: Intelligent surrogate modeling. Transp. Res. Part C: Emerg. Technol. 55, 14–23.
Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G., 2004. Short-term traffic forecasting: Overview of objectives and methods. Transp. Rev. 24 (5), 533–557.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp. Res. Part C: Emerg. Technol. 13 (3), 211–234.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2007. Spatio-temporal short-term urban traffic volume forecasting using genetically optimized modular networks. Comput.-Aid. Civ. Infrastruct. Eng. 22 (5), 317–325. https://doi.org/10.1111/j.1467-8667.2007.00488.x.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014. Short-term traffic forecasting: where we are and where we're going. Transp. Res. Part C: Emerg. Technol. 43, 3–19.
Wang, J., Chen, R., He, Z., 2019. Traffic speed prediction for urban transportation network: A path based deep learning approach. Transp. Res. Part C: Emerg. Technol. 100, 372–385.
Williams, B., 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling. Transp. Res. Rec.: J. Transp. Res. Board 1776, 194–200.
Williams, B.M., Hoel, L.A., 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. J. Transp. Eng. 129 (6), 664–672.
Wu, Y., Tan, H., 2016. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. arXiv preprint arXiv:1612.01022.
Wu, Y., Tan, H., Qin, L., Ran, B., Jiang, Z., 2018. A hybrid deep learning based traffic flow prediction method and its understanding. Transp. Res. Part C: Emerg. Technol. 90, 166–218.
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. In: International Conference on Machine Learning, pp. 2048–2057.
Yao, H., Tang, X., Wei, H., Zheng, G., Lin, Z., 2018. Revisiting spatial-temporal similarity: a deep learning framework. AAAI 2019. arXiv preprint arXiv:1604.04527.
Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X., 2017. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors 17 (7), 1501. https://doi.org/10.3390/s17071501.
Zhang, H.M., 2000. Recursive prediction of traffic conditions with neural network models. J. Transp. Eng. 126 (6), 472–481. https://doi.org/10.1061/(asce)0733-947x(2000) 126:6(472).

Zhang, Y., Zhang, Y., Haghani, A., 2014. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. Transp. Res. Part C: Emerg. Technol. 43, 65–78.

Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J., 2017. LSTM network: a deep learning approach for short-term traffic forecast. IET Intel. Transp. Syst. 11 (2), 68–75. https://doi.org/10.1049/iet-its.2016.0208.

Zheng, W., Lee, D.H., Shi, Q., 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. J. Transp. Eng. 132 (2), 114–121. https://doi.org/10.1061/(asce)0733-947x(2006) 132:2(114).

Zhu, J.Z., Cao, J.X., Zhu, Y., 2014. Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. Transp. Res. Part C: Emerg. Technol. 47, 139–154. https://doi.org/10.1016/j.trc.2014.06.011.