IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 3, 2018, accepted December 16, 2018, date of publication December 18, 2018,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2888561

STANN: A Spatio–Temporal Attentive Neural Network for Traffic Prediction

ZHIXIANG HE, CHI-YIN CHOW, (Senior Member, IEEE), AND JIA-DONG ZHANG, (Member, IEEE)
Department of Computer Science, City University of Hong Kong, Hong Kong

Corresponding authors: Chi-Yin Chow (chiychow@cityu.edu.hk) and Jia-Dong Zhang (jzhang26@cityu.edu.hk)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772445, in part by the Innovation and Technology Fund (HKSAR) under Grant UIM/334 (9440187), and in part by the City University of Hong Kong under Grant 7004684, Grant 9678132, and Grant 9680221.

ABSTRACT Recently, traffic prediction based on deep learning methods has attracted much attention. However, there still exist two major challenges, namely, dynamic spatio-temporal dependences among network-wide links and long-term traffic prediction for the next few hours. To address these two challenges, this paper proposes a spatio-temporal attentive neural network (STANN) for the network-wide and long-term traffic prediction. The STANN captures the spatial–temporal dependences based on the encoder–decoder architecture with the attention mechanisms. In the encoder, the STANN learns the spatio-temporal dependences from historical traffic series using a recurrent neural network (RNN) with long short-term memory (LSTM) units, in which a new spatial attention model is developed to consider the contribution of each link to the network-wide prediction. In the decoder, the STANN exploits another RNN with LSTM units and a temporal attention model to select the relevant and important historical spatio-temporal dependences from the encoder for long-term traffic prediction. Finally, we conduct extensive experiments to evaluate STANN on three real-world traffic datasets. The experimental results show that the STANN is significantly better than other state-of-the-art models.

INDEX TERMS Spatio-temporal data, deep neural network, attention mechanism, traffic prediction.

I. INTRODUCTION

Reliable traffic prediction models are important for supporting dynamical transportation strategies and applications in intelligent transportation systems. Techniques of traffic prediction used to be studied for decades and can be generally divided into two groups: knowledge-driven models and data-driven models. The knowledge-driven models attempt to simulate a road network and explore the performances and behaviors of the drivers in the road network [1]. In contrast, the data-driven models often estimate the future traffic by a mathematical model based on historical and current traffic data. In terms of the prediction horizon, it is generally categorized into two scales: short-term prediction (0 - 60 minutes) and long-term prediction (over 60 minutes).

Most existing works apply the data-driven models to predict traffic for links in a road network. The early data-driven models mainly include time series models (e.g., autoregressive integrated moving average model (ARIMA) [2]) and machine learning models (e.g., support vector regression (SVR) [3], and random forest (RF) [4]). With the development of deep learning techniques, deep neural networks (DNNs) have been extensively investigated for traffic prediction. For example, the work [5] uses forward neural networks to capture the non-linear properties for traffic prediction. The references [6]–[8] utilize recurrent neural networks (RNNs) that are inherently suitable for precessing time series data. The literatures [9], [10] explore the spatio-temporal dependencies of links with convolutional neural networks (CNNs). The work [11] designs a residual neural network to capture the spatio-temporal dependencies among links to predict traffic flow at next time step. The study [12] learns the temporal dependency using the encoder-decoder neural networks to predict the multi-step ahead traffic speeds.

This paper focuses on addressing two major open research challenges for traffic prediction. (1) **Dynamic spatio-temporal dependencies among network-wide links.** Some works focus on making traffic prediction for a single link, i.e., ignoring the spatio-temporal dependencies among links [5]–[8]. Other studies consider all links in the network, but they statically view all links equally important [9]–[12]. Actually, the network traffic is often determined by some

significant links or patterns which are dynamic over time. For instance, some busy links play much more significant influences on the road network and the links with free traffic have less important influences. Thus, this challenge should be addressed to achieve accurate network-wide traffic prediction. (2) **Long-term traffic prediction.** Existing traffic prediction models are not good for long-term prediction, i.e., the prediction accuracy significantly reduces with the increase of the prediction time horizon [10], [12], [13]. However, in many location-based applications, it is important to predict long-term traffic conditions for better planning and scheduling; hence, this challenge should be addressed for traffic prediction.

To address the two above-mentioned challenges, we propose a Spatio-Temporal Attentive Neural Network (STANN) based on the encoder-decoder architecture for the network-wide and long-term traffic prediction. In the encoder, STANN enhances the RNN with LSTM units by developing a spatial attention model that aims to capture the spatial dependency by considering the contribution of each link to the traffic of the whole network. In the decoder, STANN applies another RNN with LSTM units and a temporal attention model to adaptively discover the important hidden states from the encoder instead of choosing the last hidden state or simply taking the average of all hidden states. In summary, our main contributions can be stated as:

- We developed a new spatial attention model to capture the dynamic spatial dependency by considering the contribution of each link to network-wide traffic over time in the encoder.
- We designed a temporal attention model to adaptively choose important spatio-temporal information from the encoder for long-term traffic prediction in the decoder.
- To the best of our knowledge, this is the first study to utilize the encoder-decoder architecture with the spatio-temporal attention models for network-wide and long-term traffic prediction.
- We conducted extensive experiments on three real-world traffic datasets to evaluate the proposed STANN. Experimental results show that STANN significantly outperforms other state-of-the-art models.

The rest of this paper is organized as follows. Section II reviews related literatures. Section III presents our problem definition. Section IV presents the proposed STANN model for traffic prediction. Section V gives the evaluation of STANN compared with the state-of-the-art models. Finally, Section VI concludes this paper.

II. RELATED WORK

This section briefly reviews the related work about traffic prediction and attention mechanism.

A. NON-DEEP LEARNING MODELS

Non-deep learning based models mainly include knowledge-driven methods, time series models and machine learning models. Knowledge-driven models usually apply queuing theory and present the dynamical traffic description of the road network, e.g., visual interactive system for transport algorithms [14], transportation analysis simulation system [15] and tools for operational planning of transportation networks [16]. Time series models focus on discovering the patterns of the temporal variation of traffic data for prediction, e.g., ARIMA [2] and Kalman filter [17], [18]. However, they only depend on the traffic sequential data and ignore the dynamic spatial dependency. Machine learning models predict the traffic based on the similarities between current data and historical data, e.g., K-nearest neighbors (KNN) [19], RF [4], and SVR [3], which achieve much better prediction accuracy. However, they perform badly for long-term prediction, especially on complex networks.

B. DEEP LEARNING BASED MODELS

DNNs are applied to traffic prediction due to their great abilities of processing non-linear properties. For example, the work [5] uses forward neural networks for traffic prediction. An object-oriented dynamic neural network model is designed for short-term traffic conditions prediction [20]. The study [21] proposes a neural network model with a multi-layer structural optimization strategy for traffic flow prediction. The theory of conditional probability and Bayes' rule are used to integrate the predictions from single neural networks as the final traffic flow prediction [22].

Especially, RNNs are adopted for traffic prediction because they are inherently suitable for processing time series data [23]. However, RNNs suffer from the vanishing gradient problem that prevents them from learning the long-term dependency. Some variants of RNNs, such as the RNN with LSTM units [24] or gated recurrent units (GRU) [13], alleviate this problem using a gated mechanism. For example, the study [6] uses LSTM for traffic speed prediction by using the speed data from traffic microwave detectors. The research [7] develops a three-layer LSTM to predict travel times of highway links. The work [8] optimizes the structure design and hyperparameter settings of deep learning models for traffic prediction. Unfortunately, these studies only focus on a single link and fail to consider the spatial correlations of links over the road network.

There are also some studies considering the spatio-temporal dependencies for network-wide traffic prediction. For instance, the study [25] utilizes a deep belief network for traffic prediction. A stacked auto-encoder model is proposed to capture spatio-temporal correlations from the traffic flow data [9]. The work [9] explores the spatial dependency with CNNs over an image which is converted from the network traffic. A Hetero-ConvLSTM model is developed for traffic accident prediction on heterogeneous spatio-temporal data [26]. STGCN [10] utilizes CNNs over graph structured traffic data to capture the spatio-temporal dependencies. A residual network is designed to capture spatio-temporal dependencies for traffic flow prediction [11]. DCRNN [12] applies bidirectional random walks on the graph to capture the spatial dependency and the encoder-decoder architecture

TABLE 1. The important notations used in this paper.

Symbol	Description
$\mathtt{N}$	number of links of the network
$\mathtt{T}$, $\mathtt{T}'$	current time step, number of future time steps
t, t'	past time step t, future time step t' from $\mathtt{T}$
$\mathbf{x}^i \in \mathbb{R}^{\mathtt{T}}$	historical traffic speeds of link i over past $\mathtt{T}$ steps
$\mathbf{x}_t \in \mathbb{R}^{\mathtt{N}}$	traffic speed vector of all $\mathtt{N}$ links at t
$\mathbf{X}_{\mathtt{T}}^{\mathtt{N}} \in \mathbb{R}^{\mathtt{N}\times\mathtt{T}}$	historical traffic speed tensor of $\mathtt{N}$ links over past $\mathtt{T}$ steps
$\hat{\mathbf{x}}_{t'} \in \mathbb{R}^{\mathtt{T}'}$	predicted traffic speeds of all $\mathtt{N}$ links at t'
$\hat{\mathbf{X}}_{\mathtt{T}'}^{\mathtt{N}} \in \mathbb{R}^{\mathtt{N}\times\mathtt{T}'}$	predicted traffic speeds of all $\mathtt{N}$ links for future $\mathtt{T}'$ steps
$\mathtt{M}$, $\mathtt{L}$	dimensions of the spatial attention model and encoder LSTM
$\mathbf{v}_s, \mathbf{b}_s \in \mathbb{R}^{\mathtt{L}}$, $\mathbf{W}_s \in \mathbb{R}^{\mathtt{L}\times 2\mathtt{M}}$, $\mathbf{U}_s \in \mathbb{R}^{\mathtt{L}\times\mathtt{T}}$	parameters in spatial attention model
a_t^i, α_t^i	spatial weight and normalized spatial weight of link i at t
$\mathbf{e}_t \in \mathbb{R}^{\mathtt{N}}$	spatial attention vector at t
$\mathbf{f}_t$, $\mathbf{i}_t$, $\mathbf{o}_t$	output of forget gate, input gate and output gate in the encoder LSTMs at t
$\mathbf{c}_t, \mathbf{h}_t \in \mathbb{R}^{\mathtt{M}}$	cell state, spatio-temporal hidden state in encoder at t
$\mathtt{P}$, $\mathtt{Q}$	dimensions of the decoder LSTM and the temporal attention model
$\mathbf{v}_d, \mathbf{b}_d \in \mathbb{R}^{\mathtt{Q}}$, $\mathbf{W}_d \in \mathbb{R}^{\mathtt{Q}\times 2\mathtt{P}}$, $\mathbf{U}_d \in \mathbb{R}^{\mathtt{Q}\times\mathtt{M}}$	parameters in temporal attention model
$d_t'^i$, β_t^i	temporal weight, normalized temporal weight of link i at t'
$\mathbf{y}_{t'} \in \mathbb{R}^{\mathtt{M}}$	temporal context at t'
$\mathbf{c}'_{t'}, \mathbf{h}'_{t'} \in \mathbb{R}^{\mathtt{P}}$	cell state, spatio-temporal hidden state in the decoder LSTMs at t'
$\mathbf{W}_x \in \mathbb{R}^{\mathtt{N}\times(\mathtt{M}+\mathtt{P})}$, $\mathbf{b}_x \in \mathbb{R}^{\mathtt{N}}$	parameters in fully connected layer

to capture the temporal dependency. However, these studies equally consider all links in the network and some of them use CNNs that perform poorly for long-term prediction. To address them, our STANN applies a spatial attention model to consider the contribution of each link with respect to the whole network and a temporal attention model to select the important spatio-temporal states from historical series data.

C. ATTENTION MECHANISM

The encoder-decoder network is often used for sequence prediction and consists of an encoder to map inputs to hidden states and a decoder to decode the hidden states for making prediction [13]. Nonetheless, its performance will degrade rapidly with the increasing length of the inputs [13]. To handle this problem, the attention mechanism has been proposed for choosing important historical information and firstly applied in the encoder-decoder network for machine translation [27]. A few existing studies [12], [28] apply the encoder-decoder network without the attention mechanism for traffic prediction, so their performances greatly degrade with the increasing length of the inputs. Although the work [29] considers the attention mechanism, it only predicts the traffic of hotspots according to map queries. In other words, it only focuses on the traffic of the links that are near the query destinations or events. To this end, in this paper we enhance the encoder-decoder architecture for traffic prediction, by developing a spatial attention model over network-wide links to capture spatio-temporal dependencies in the encoder and designing a temporal attention model to catch the long-term spatio-temporal dependencies in the decoder.

III. PROBLEM DEFINITION

The key notations used in this paper are described in TABLE 1. This section presents the preliminaries and the studied problem.

A. PRELIMINARIES

A road network consists of $\mathtt{N}$ links, in which links are connected by intersections. Each link generates a time series of traffic speeds, represented by a row vector,

$$\mathbf{x}^i = (x_1^i, \ldots, x_t^i, \ldots, x_{\mathtt{T}}^i) \in \mathbb{R}^{\mathtt{T}}, \tag{1}$$

where x_t^i denotes the traffic speed of link i during time step t, and $\mathtt{T}$ denotes the current time step. The time series vectors of all $\mathtt{N}$ links are concatenated along the first (i.e., row) dimension into the speed matrix,

$$\mathbf{X}_{\mathtt{T}}^{\mathtt{N}} = (\mathbf{x}^1; \mathbf{x}^2; \ldots; \mathbf{x}^{\mathtt{N}})^\top \in \mathbb{R}^{\mathtt{N}\times\mathtt{T}}. \tag{2}$$

The speed matrix can be also splitted into speed columns, written as

$$\mathbf{X}_{\mathtt{T}}^{\mathtt{N}} = (\mathbf{x}_1, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_{\mathtt{T}}) \in \mathbb{R}^{\mathtt{N}\times\mathtt{T}}, \tag{3}$$

where each column $\mathbf{x}_t \in \mathbb{R}^{\mathtt{N}}$ denotes the traffic speeds during time step t of all $\mathtt{N}$ links in the road network.

B. STUDIED PROBLEM

Given the historical speed matrix $\mathbf{X}_{\mathtt{T}}^{\mathtt{N}}$ for the whole road network consisting of $\mathtt{N}$ links by the current time step $\mathtt{T}$, the goal

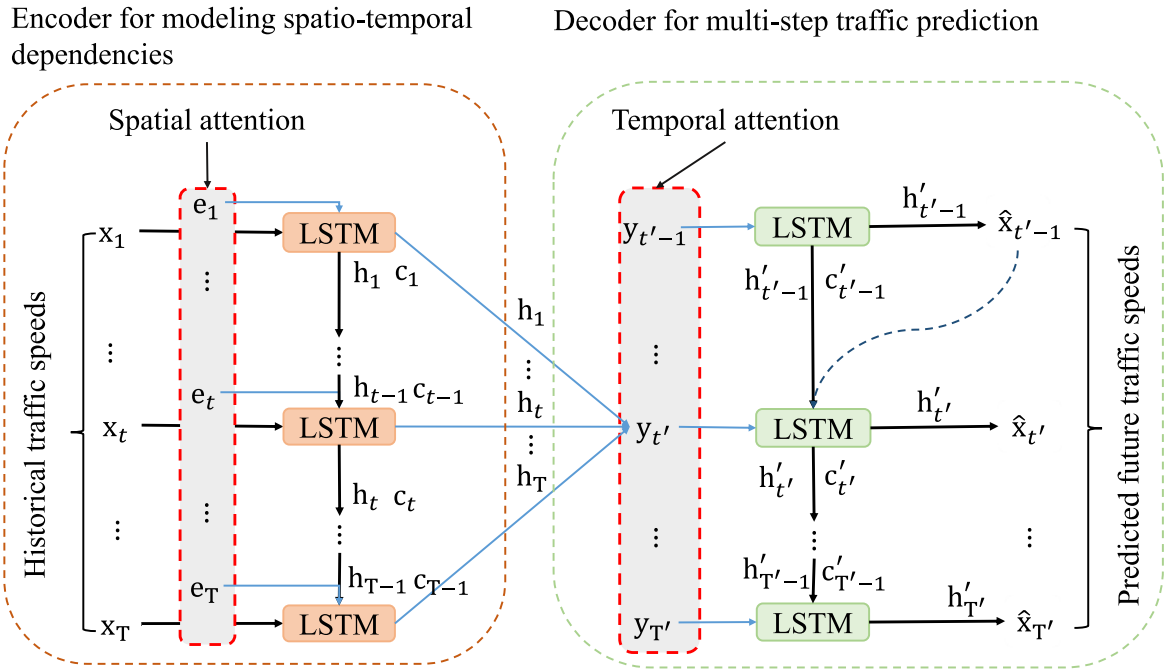

FIGURE 1. The architecture of STANN with two components: the encoder for modeling spatio-temporal dependencies and the decoder for multi-step traffic prediction. $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T$ are the output of the spatial attention model, see FIGURE 2; $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{T'}$ are the outputs of the temporal attention model, see FIGURE 3; $\mathbf{h}_1, \mathbf{h}_t, \ldots, \mathbf{h}_T$ and $\mathbf{h}'_1, \mathbf{h}'_2, \ldots, \mathbf{h}'_{T'}$ are the spatio-temporal hidden states in the encoder and decoder, respectively; $\mathbf{c}_1, \mathbf{c}_t, \ldots, \mathbf{c}_T$ and $\mathbf{c}'_1, \mathbf{c}'_2, \ldots, \mathbf{c}'_{T'}$ are the cell states in the encoder and decoder, respectively.

is to predict the network-wide traffic speeds over the next T' time steps, denoted as $\hat{\mathbf{X}}^N_{T'} = (\hat{\mathbf{x}}_{T+1}, \ldots, \hat{\mathbf{x}}_{T+t'}, \ldots, \hat{\mathbf{x}}_{T+T'}) \in \mathbb{R}^{N\times T'}$, or $\hat{\mathbf{X}}^N_{T'} = (\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{t'}, \ldots, \hat{\mathbf{x}}_{T'}) \in \mathbb{R}^{N\times T'}$ for short, where $\hat{\mathbf{x}}_{t'} \in \mathbb{R}^N$ is the predicted speeds for all links at the time step $T + t'$, i.e., the t'-th time step after T.

IV. SPATIAL-TEMPORAL ATTENTIVE NEURAL NETWORK (STANN)

At first, this section describes the architecture of the proposed STANN model in Section IV-A. Then we present the encoder for modeling spatio-temporal dependencies and the decoder for multi-step traffic prediction in Sections IV-B and IV-C, respectively. Finally, the training procedure of STANN is given in Section IV-D.

A. ARCHITECTURE OF STANN

Our proposed STANN applies the encoder-decoder architecture, as shown in FIGURE 1. STANN consists of two main components. (1) **Encoder for modeling spatio-temporal dependencies.** The encoder applies a RNN with LSTM units to capture the spatio-temporal dependencies from historical traffic time series, as shown in the left part of FIGURE 1. Compared to the existing works [10], [12], [30], we devise a spatial attention model for the spatial dependency by considering the spatial attention weight vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T$ of all links to the network-wide traffic. The weight vector $\mathbf{e}_t$ at time step t is generated by the multilayer perceptron with the last hidden state $\mathbf{h}_{t-1}$ and cell state $\mathbf{c}_{t-1}$ in the encoder LSTM, see FIGURE 2. The spatio-temporal hidden states $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T)$ at all historical time steps are learned from the encoder LSTM with the spatial attention weight vectors. These hidden states serve as the inputs of the decoder component for traffic prediction at future time steps. (2) **Decoder for multi-step traffic prediction.** The main difference of our STANN from the previous works [10], [12], [30] is that a temporal attention model over the historical spatio-temporal hidden states $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T)$ is embedded into the decoder that utilizes another RNN with LSTM units, as shown in the right part of FIGURE 1. The temporal attention model for long-term prediction provides the temporal context $\mathbf{y}_{t'}$ which is a weighted combination of the spatio-temporal hidden states $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T)$ from the encoder. The temporal

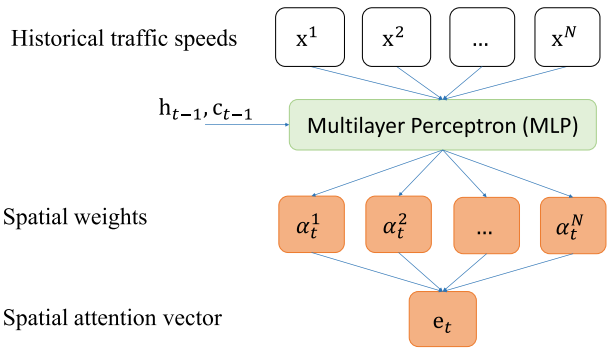

FIGURE 2. The spatial attention model in the encoder.

attention weights are calculated by the multilayer perceptron with the last hidden state $\mathbf{h}'_{t-1}$ and cell state $\mathbf{c}'_{t-1}$ in the decoder LSTM, see FIGURE 3. In other words, the decoder learns the future spatio-temporal states $(\mathbf{h}'_1, \mathbf{h}'_2, \ldots, \mathbf{h}'_{\mathtt{T}'})$ from the relevant states $(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{\mathtt{T}})$ to predict the future traffic $\hat{\mathbf{X}}^{\mathtt{N}}_{\mathtt{T}'}$ for future $\mathtt{T}'$ time steps.

B. ENCODER FOR MODELING SPATIO-TEMPORAL DEPENDENCIES

This study aims to predict the network-wide traffic speeds for multiple future time steps. A simple method is to make speed prediction for each link independently [7], [8] or consider all links with equal weights [12]. In fact, these links play different roles and their roles also change over time. For example, at a time step some links have static traffic conditions whereas other links show dynamic traffic conditions, but at another time step these traffic conditions may vary reversely. Moreover, these links show spatio-temporal dependencies. For instance, the traffic speeds of upstream links rely on the traffic conditions of downstream links, and the downstream links propagate congestions to the upstream links quickly.

Therefore, we consider the traffic conditions of all links in the road network as a whole, and enhance the conventional LSTM that is originally designed for learning temporal dependency, by developing a spatial attention model in order to simultaneously capture the spatio-temporal dependencies over time, as depicted in FIGURE 2. Specifically, given the previous cell state $\mathbf{c}_{t-1} \in \mathbb{R}^{\mathtt{M}}$ and hidden state $\mathbf{h}_{t-1} \in \mathbb{R}^{\mathtt{M}}$ of LSTM for all links, the spatial attention model learns weight of contribution of link i to the traffic of network-wide links at t by the multilayer perceptron (MLP):

$$a_t^i = \mathbf{v}_s^\top \tanh(\mathbf{W}_s[\mathbf{c}_{t-1}; \mathbf{h}_{t-1}] + \mathbf{U}_s(\mathbf{x}^i)^\top + \mathbf{b}_s), \tag{4}$$

where $\mathbf{x}^i \in \mathbb{R}^{\mathtt{T}}$ is the traffic speed row vector of link i, $\mathbf{v}_s \in \mathbb{R}^{\mathtt{L}}$, $\mathbf{b}_s \in \mathbb{R}^{\mathtt{L}}$, $\mathbf{W}_s \in \mathbb{R}^{\mathtt{L}\times 2\mathtt{M}}$ and $\mathbf{U}_s \in \mathbb{R}^{\mathtt{L}\times \mathtt{T}}$ are the model parameters, $\mathtt{L}$ and $\mathtt{M}$ are the dimensions of the spatial attention model and encoder LSTM, respectively. The obtained attention a_t^i adaptively catches the importance of link i to the traffic of the network. The spatial attention weights of all links are normalized into [0, 1] using a softmax function to make the sum of all attention weights equal to one:

$$\alpha_t^i = \frac{\exp(a_t^i)}{\Sigma_{j=1}^{\mathtt{N}} \exp(a_t^j)}. \tag{5}$$

Then, all spatial attention weights are concatenated as a column vector:

$$\mathbf{e}_t = (\alpha_t^1, \alpha_t^2, \ldots, \alpha_t^{\mathtt{N}})^\top, \tag{6}$$

which implies the spatial dependencies on significant traffic links at time step t. For example, if there are congestions in the upstream links or the downstream links, it will pay more attention to such links since they make large impacts on the network-wide traffic.

The enhanced LSTM generates next cell state $\mathbf{c}_t \in \mathbb{R}^{\mathtt{M}}$ and hidden state $\mathbf{h}_t \in \mathbb{R}^{\mathtt{M}}$ by the following steps [24]:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}; \mathbf{x}_t; \mathbf{e}_t] + \mathbf{b}_f), \tag{7}$$
$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}; \mathbf{x}_t; \mathbf{e}_t] + \mathbf{b}_i), \tag{8}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}; \mathbf{x}_t; \mathbf{e}_t] + \mathbf{b}_o), \tag{9}$$
$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}; \mathbf{x}_t; \mathbf{e}_t] + \mathbf{b}_c), \tag{10}$$
$$\mathbf{c}_t = \mathbf{f}_t \otimes \mathbf{c}_{t-1} + \mathbf{i}_t \otimes \hat{\mathbf{c}}_t, \tag{11}$$
$$\mathbf{h}_t = \mathbf{o}_t \otimes \tanh(\mathbf{c}_t), \tag{12}$$

where $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$, $\mathbf{W}_c \in \mathbb{R}^{\mathtt{M}\times(\mathtt{M}+2\mathtt{N})}$ and $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_o$, $\mathbf{b}_c \in \mathbb{R}^{\mathtt{M}}$ are the parameters of the forget gate, input gate, output gate and memory cell, respectively, σ is the sigmoid function, tanh is the hyperbolic tangent function, $\otimes$ denotes the element-wise multiplication, $\mathtt{M}$ is the dimension of LSTM, and $\mathtt{N}$ is the number of links in the road network.

It is worth emphasizing that: (1) The improved LSTM still utilizes forget gate, input gate, output gate and memory cell to control the passing of information at each time step and catch the long-term temporal dependencies in the traffic speed time series of all links. (2) The LSTM also exploits the spatial attention model to capture the spatial dependencies at the same time. (3) The LSTM outputs a sequence of hidden states $(\mathbf{h}_1, \ldots, \mathbf{h}_t, \ldots, \mathbf{h}_{\mathtt{T}})$ that have encoded the spatio-temporal dependencies at different time steps and are used for multi-step traffic prediction, as presented in Section IV-C.

C. DECODER FOR MULTI-STEP TRAFFIC PREDICTION

To predict the traffic speeds of multiple future time steps, the traditional decoder with LSTM [13] can be used to decode the hidden state sequence $(\mathbf{h}_1, \ldots, \mathbf{h}_t, \ldots, \mathbf{h}_{\mathtt{T}})$ into a fixed-length target sequence. Unfortunately, its performance will degrade rapidly with the increasing length of the inputs. Therefore, this paper improves the decoder with a temporal attention model, as depicted FIGURE 3. It can adaptively discover the important hidden states throughout all previous time steps instead of choosing the last hidden state or simply taking the average of all hidden states. In other words, the attention model learns the weights of all hidden states and retrieves

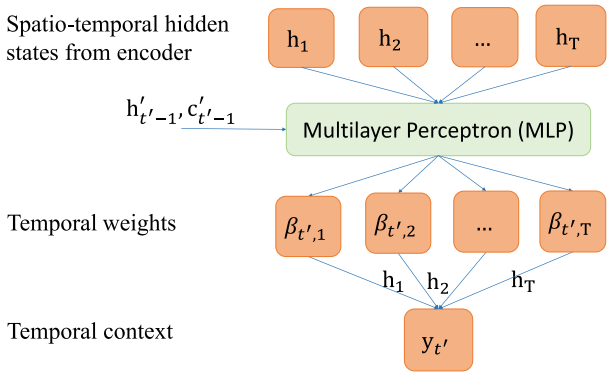

FIGURE 3. The temporal attention model in the decoder.

their weighted sum to capture the dynamical temporal correlations between future and historical time steps.

Specifically, given the previous cell state $\mathbf{c}'_{t'-1} \in \mathbb{R}^{\mathtt{P}}$ and hidden state $\mathbf{h}'_{t'-1} \in \mathbb{R}^{\mathtt{P}}$ of the LSTM in the decoder, the traffic correlation between the t'-th future time step and the t-th historical hidden state $\mathbf{h}_t \in \mathbb{R}^{\mathtt{M}}$ is measured by multilayer perceptron,

$$d_{t',t} = \mathbf{v}_d^{\top} \tanh(\mathbf{W}_d[\mathbf{c}'_{t'-1}; \mathbf{h}'_{t'-1}] + \mathbf{U}_d\mathbf{h}_t + \mathbf{b}_d), \tag{13}$$

where $\mathbf{v}_d \in \mathbb{R}^{\mathtt{Q}}$, $\mathbf{b}_d \in \mathbb{R}^{\mathtt{Q}}$, $\mathbf{W}_d \in \mathbb{R}^{\mathtt{Q}\times 2\mathtt{P}}$, and $\mathbf{U}_d \in \mathbb{R}^{\mathtt{Q}\times\mathtt{M}}$ are the model parameters, $\mathtt{P}$ and $\mathtt{Q}$ are the dimensions of the LSTM decoder and the temporal attention model, respectively. Similarly, the attention weights of all historical hidden states are normalized into [0, 1] using a soft-max function to make all weights sum to one:

$$\beta_{t',t} = \frac{\exp(d_{t',t})}{\sum_{t''=1}^{\mathtt{T}} \exp(d_{t',t''})}. \tag{14}$$

Then the temporal context vector $\mathbf{y}_{t'}$ is calculated by the weighted sum of all historical hidden states, given by

$$\mathbf{y}_{t'} = \sum_{t=1}^{\mathtt{T}} \beta_{t',t}\mathbf{h}_t. \tag{15}$$

With the temporal attention model, we get the temporal context vector $\mathbf{y}_{t'}$ at the future time step t'. $\mathbf{y}_{t'}$ is further concatenated with the last output of the decoder $\hat{\mathbf{x}}_{t'-1}$ to update the hidden state $\mathbf{h}'_{t'}$ of the decoder, concisely written as:

$$\mathbf{h}'_{t'} = LSTM_{decoder}(\mathbf{h}'_{t'-1}, [\hat{\mathbf{x}}_{t'-1}; \mathbf{y}_{t'}]). \tag{16}$$

Note that the update steps are similar to Equations (7) to (12). Finally, the context vector $\mathbf{y}_{t'} \in \mathbb{R}^{\mathtt{M}}$ and hidden state $\mathbf{h}'_{t'} \in \mathbb{R}^{\mathtt{P}}$ are concatenated to predict the traffic speeds $\hat{\mathbf{x}}_{t'} \in \mathbb{R}^{\mathtt{N}}$ of all links at the future time step t', based on a fully connected layer (FCL), given by,

$$\hat{\mathbf{x}}_{t'} = ReLu(\mathbf{W}_x[\mathbf{y}_{t'}; \mathbf{h}'_{t'}] + \mathbf{b}_x), \tag{17}$$

where $ReLu$ is the activation function [31], $\mathbf{W}_x \in \mathbb{R}^{\mathtt{N}\times(\mathtt{M}+\mathtt{P})}$ and $\mathbf{b}_x \in \mathbb{R}^{\mathtt{N}}$ are the model parameters.

D. TRAINING

We here present the training procedure of the proposed STANN. The pseudo-code of STANN is presented in Algorithm 1. The model is trained end to end. We utilize the Adam optimization algorithm [32] to minimize the loss function of Mean Square Error (MSE) for the predicted sequence $\hat{\mathbf{X}}^{\mathtt{N}}_{\mathtt{T}'} = (\hat{\mathbf{x}}_{\mathtt{T}+1}, \ldots, \hat{\mathbf{x}}_{\mathtt{T}+t'}, \ldots, \hat{\mathbf{x}}_{\mathtt{T}+\mathtt{T}'}) \in \mathbb{R}^{\mathtt{N}\times\mathtt{T}'}$ and the true sequence $\mathbf{X}^{\mathtt{N}}_{\mathtt{T}'} = (\mathbf{x}_{\mathtt{T}+1}, \ldots, \mathbf{x}_{\mathtt{T}+t'}, \ldots, \mathbf{x}_{\mathtt{T}+\mathtt{T}'}) \in \mathbb{R}^{\mathtt{N}\times\mathtt{T}'}$, on the road network.

$$Loss(\Theta) = \frac{1}{\mathtt{N}\mathtt{T}'}\|\hat{\mathbf{X}}^{\mathtt{N}}_{\mathtt{T}'} - \mathbf{X}^{\mathtt{N}}_{\mathtt{T}'}\|^2, \tag{18}$$

where Θ represents the set of model parameters to be learned in the training procedure, including all parameters in LSTM, $\mathbf{v}_s \in \mathbb{R}^{\mathtt{L}}$, $\mathbf{b}_s \in \mathbb{R}^{\mathtt{L}}$, $\mathbf{W}_s \in \mathbb{R}^{\mathtt{L}\times 2\mathtt{M}}$ and $\mathbf{U}_s \in \mathbb{R}^{\mathtt{L}\times\mathtt{T}}$ in the spatial attention model (Equation (4)), $\mathbf{v}_d \in \mathbb{R}^{\mathtt{Q}}$, $\mathbf{b}_d \in \mathbb{R}^{\mathtt{Q}}$, $\mathbf{W}_d \in \mathbb{R}^{\mathtt{Q}\times 2\mathtt{P}}$, and $\mathbf{U}_d \in \mathbb{R}^{\mathtt{Q}\times\mathtt{M}}$ in the temporal attention model (Equation (13)) and $\mathbf{W}_x \in \mathbb{R}^{\mathtt{N}\times(\mathtt{M}+\mathtt{P})}$ and $\mathbf{b}_x \in \mathbb{R}^{\mathtt{N}}$ in the fully connected layer (Equation (17)) for final prediction.

Algorithm 1 Pseudo-Code for Training Procedure of STANN

Input: Training data including historical data $\mathbf{X}^{\mathtt{N}}_{\mathtt{T}}$ and future ground truth $\mathbf{X}^{\mathtt{N}}_{\mathtt{T}'}$, all hyperparameters;
Output: Learned STANN model;
Initialization: All training parameters Θ in STANN;
for *each epoch* **do**
  Shuffle training data;
  for *each batch of* $\mathbf{X}^{\mathtt{N}}_{\mathtt{T}}$ *in training data* **do**
    // Encoder for modeling spatio-temporal dependencies from historical data;
    for $t = 0$ *to* $\mathtt{T}$ **do**
      Spatial attention model $\rightarrow \mathbf{e}_t$ in Equation (6);
      Encoder LSTM units with $\mathbf{x}_t$ and $\mathbf{e}_t \rightarrow \mathbf{h}_t$, $\mathbf{c}_t$;
    end
    // Decoder for multi-step traffic prediction;
    Initial decoder hidden state: $\mathbf{h}'_0 = \mathbf{c}_{\mathtt{T}}$;
    for $t' = 0$ *to* $\mathtt{T}'$ **do**
      Temporal attention model $\rightarrow \mathbf{y}_{t'}$ in Equation (15);
      Decoder LSTM units with $\mathbf{x}_{t'-1}$ and $\mathbf{y}_{t'} \rightarrow \mathbf{h}'_{t'}$;
      A FCL with $\mathbf{h}'_{t'}$ and $\mathbf{y}_{t'} \rightarrow \hat{\mathbf{x}}_{\mathtt{t}'}$;
    end
    Optimize Θ by minimizing Equation (18);
  end
end

V. EXPERIMENTS

We first describe the experimental settings in Section V-A and the models compared with STANN in Section V-B. Experimental results are presented and analyzed in Sections V-C, V-D, V-E, and V-F.

A. EXPERIMENTAL SETTINGS

1) DATASETS

In this study, we conduct experiments over three traffic datasets (HK-KL, ST, TM) in Hong Kong to evaluate the performance of STANN. Hong Kong consists of three main regions, Hong Kong Island (HK), Kowloon Peninsula (KL), and New Territories (NT), as shown in FIGURE 4. We collected the traffic data from the Traffic Speed Map system (TSM) supported by the transportation department of Hong Kong.[1] TSM provides the traffic speeds of major routes and urban roads in the three regions. There are 605 specific links with real-time traffic speeds, in which there are

[1] http://resource.data.one.gov.hk/td/speedmap.xml

FIGURE 4. The three main regions (i.e., Hong Kong Island, Kowloon Peninsula, and New Territories) in Hong Kong.

440 links (speed limit larger than 50 kilometer per hour) and 165 links (speed limit less than 50 kilometers per hour). The traffic speed is calculated by an on-line estimator using three types real-time data (auto-toll tag data, GPS data, video image processing data) and an off-line estimator using annual statistical traffic data [33], [34]. The traffic speed of each link is updated every two minutes. Since the traffic speeds are from urban areas, the complex traffic environment would have a great impact on the traffic speed, which may make the traffic speed varies a lot. To make the traffic speed predictable, we sample the traffic speed per 10 minutes using the average speed for each link.

FIGURE 5. Road networks of three datasets in Hong Kong.

The dataset HK-KL contains the traffic data in HK and KL. As Sha Tin (ST) and Tuen Mun (TM) are located in NT, datasets ST and TM denote the traffic data in ST and TM, respectively. Note that we consider them as three datasets because there are no available traffic speed information on the links connecting them, as depicted in FIGURE 5. The dataset details are shown in TABLE 2. Each dataset is segmented into three parts based on the timestamp: the first 70% for training, the next 20% for validation, and the remaining 10% for testing.

TABLE 2. Details of datasets.

Dataset	HK-KL	ST	TM
Number of links (N)	202	198	205
Total links	605		
Time spans	1/1/2017 - 6/30/2018		
Time interval	10 minutes		
Data instances	78,614		

2) EVALUATION METRICS

To evaluate the performance of traffic prediction, we use three standard metrics including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Their definitions are as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\|\hat{x}_i - x_i\|_1, \tag{19}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2}, \tag{20}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{\hat{x}_i - x_i}{x_i}| \times 100, \tag{21}$$

where N is the size of the testing set, x_i and $\hat{x}_i$ are the ground truth and the predicted value, respectively. For these three metrics, smaller values indicate better performance.

3) HYPERPARAMETER SETTINGS

We set the length of window $\mathtt{T}$ to 12 steps (i.e., $\mathtt{T} = 12$), namely, the traffic conditions of past 120 minutes are used to predict the traffic conditions in the next 30, 60, 90 and 120 minutes ($\mathtt{T}' = 3, 6, 9, 12$). To capture the spatio-temporal dependencies of the whole road network, we set the size of hidden states to the number of links for each dataset as shown in TABLE 2, i.e., $\mathtt{M} = \mathtt{N}$. The number of LSTM layers in both the encoder and decoder is set to 2, where we also set $\mathtt{P} = \mathtt{N}$ and $\mathtt{L} = \mathtt{Q} = \mathtt{T}$. We train the model using the Adam optimization algorithm [32], in which we set the learning rate to 0.001, dropout rate to 0.1, batch size to 128 and epoch to 100.

B. COMPARED MODELS

We compare STANN with the following models, and tuned the parameters for them.

- SVR [3]: SVR is a well-known machine learning method. We use Radial Basis Function kernel for training, in which the kernel coefficient is set to 0.1.
- RF [4]: RF is an ensemble learning method. We use it for regression to predict the future traffic. In the RF model, 10 trees without maximum depth are built. The minimum number of samples required to split an internal node is 128 and random state is 2.

TABLE 3. Performance comparison of different models on dataset ST.

Time	30 minutes			60 minutes			90 minutes			120 minutes		
Model	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	7.30	5.76	12.03	8.32	6.38	14.38	8.97	6.75	16.16	9.25	6.91	16.99
RF	5.53	3.26	7.72	6.65	3.93	10.17	7.40	4.42	12.06	7.77	4.74	13.09
Seq2seq	5.64	3.49	8.15	5.86	3.55	8.39	5.99	3.58	8.62	6.19	3.64	9.05
STGCN	4.90	**2.81**	6.30	5.93	3.22	7.56	6.63	3.49	8.60	7.38	3.86	9.58
DCRNN	5.73	2.94	7.25	6.94	3.39	9.25	7.89	3.76	11.24	8.64	4.07	12.86
STANN	**4.40**	2.85	**6.14**	**4.47**	**2.88**	**6.19**	**4.53**	**2.90**	**6.29**	**4.77**	**2.99**	**6.62**

TABLE 4. Performance comparison of different models on dataset TM.

Time	30 minutes			60 minutes			90 minutes			120 minutes		
Model	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	6.10	4.93	9.13	6.64	5.27	10.13	6.96	5.47	10.77	7.09	5.55	11.02
RF	4.52	2.93	5.96	5.14	3.32	7.00	5.54	3.60	7.76	5.75	3.77	8.20
Seq2seq	4.80	3.17	6.37	4.93	3.19	6.51	5.03	3.23	6.64	5.29	3.40	6.92
STGCN	4.22	2.66	5.28	4.77	2.90	5.92	5.13	3.07	6.33	5.52	3.23	6.79
DCRNN	4.67	2.89	5.95	5.18	3.08	6.52	5.47	3.20	6.85	5.64	3.29	7.03
STANN	**3.94**	**2.64**	**5.13**	**3.97**	**2.65**	**5.15**	**4.01**	**2.68**	**5.21**	**4.15**	**2.75**	**5.39**

TABLE 5. Performance comparison of different models on dataset HK-KL.

Time	30 minutes			60 minutes			90 minutes			120 minutes		
Model	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	6.02	4.79	15.75	7.31	5.75	19.97	8.18	6.42	23.32	8.75	6.90	25.73
RF	4.72	3.12	10.13	5.95	4.00	14.04	6.75	4.61	17.01	7.23	5.03	19.06
Seq2seq	4.93	3.28	11.42	5.20	3.35	11.80	5.32	3.39	11.96	5.44	3.46	12.19
STGCN	4.33	2.62	8.66	5.22	3.01	10.36	5.70	3.24	11.40	6.00	3.39	12.12
DCRNN	4.46	2.62	8.75	4.81	2.73	9.35	4.93	2.78	9.59	5.09	2.87	9.96
STANN	**3.58**	**2.46**	**7.99**	**3.57**	**2.44**	**7.91**	**3.63**	**2.47**	**8.01**	**3.89**	**2.60**	**8.53**

- Seq2seq [30]: SeqSeq is also an encoder-decoder network which uses a RNN as the encoder to map the input sequences into hidden states and another RNN as the decoder to decode the hidden states for prediction. Both the encoder and decoder are two LSTM layers in the experiments; in each LSTM layer, the number of the neural units is the number of links in the road network, the learning rate is 0.0001, and epoch is 70.
- STGCN [10]: A deep learning framework for traffic prediction with spatio-temporal graph convolutional neural networks. The channels of the three layers in ST-Conv block are 64, 16, and 64. The kernel size in both the graph convolution and temporal convolution is set to 3. STGCN is trained by minimizing the MSE using RMSprop optimizer for 50 epochs with batch size as 50. The initial learning rate is 0.001 with a decay rate of 0.7 after every 5 epochs.
- DCRNN [12]: A deep learning framework for traffic prediction with diffusion convolutional recurrent neural network. DCRNN is trained by minimizing the MAE with Adam optimizer. In both the encoder and decoder, there are two LSTM layers, each with 64 units. The initial learning rate is 0.01, and then it is reduced to 10% every 10 epochs starting at the 20th epoch. The size of convolution kernel is set to 3. The batch size is set to 64. The maximum epoch is 100 and the early stop is applied by monitoring the validation error.

We use Scikit-Learn library[2] to implement SVR and RF in Python. Since SVR and RF can only predict the traffic speed for a single link we train a model for each link and report their average performances on all links. Besides, all compared deep learning models are implemented by TensorFlow which is an open source library.[3]

C. OVERALL COMPARISON

TABLE 3, 4, and 5 present the results of all evaluated models for predicting the traffic speeds of next 30, 60, 90, and 120 minutes on the three datasets. In general, deep learning

[2]https://scikit-learn.org
[3]https://www.tensorflow.org

TABLE 6. Comparison for STANN, SANN and TANN on dataset HK-KL.

Time	30 minutes			60 minutes			90 minutes			120 minutes		
Model	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SANN	3.81	2.61	8.56	4.18	2.77	9.26	4.29	2.83	9.54	4.56	2.97	10.15
TANN	3.8	2.57	8.47	4.13	2.76	9.32	4.21	2.79	9.42	4.48	2.92	9.91
STANN	**3.58**	**2.46**	**7.99**	**3.57**	**2.44**	**7.91**	**3.63**	**2.47**	**8.01**	**3.89**	**2.60**	**8.53**

based models, including Seq2seq, DCRNN, and STGCN outperform traditional machine learning models, including SVR and RF, which emphasizes the role of temporal dependency. More importantly, STANN achieves the best performance in terms of all the metrics for all prediction horizons, in all three datasets, especially for long-term prediction. This implies the effectiveness of our STANN with the spatial and temporal attention mechanisms.

Following, we analyze the results in details: (1) On dataset ST in TABLE 3, for 30-minute ahead prediction, although DCRNN and STGCN are very close to our STANN according to MAE, STANN improves the performance by 10.2% in terms of RMSE and MAPE. For long-term horizon 120 minutes, STANN records the improvement of at least 17.86% on MAE and 22.94% on RMSE compared to other evaluated models. (2) On dataset TM in TABLE 4, at 30 minutes, STGCN achieves a very close MAE to STANN but its performance is worse than our STANN. At 120 minutes, the improvements of STANN are about 14.86% on MAE and 24.81% on RMSE in comparison with all other models. (3) Compared to ST and TM, HK-KL is much more challenging due to its complex environments (HK and KL are located in the urban center which have complicated traffic conditions). Specifically, on dataset HK-KL in TABLE 5, for 30-minute ahead prediction, STANN makes 6.11% and 17.32% improvements against the second best STGCN on MAE and RMSE, respectively; for 120-minute ahead prediction, there are 11.26% improvements on MAE and 25.48% on RMSE over DCRNN. (4) The advantage of STANN becomes more clearer with the increase of the prediction horizon, which verifies the advantage of STANN with attention mechanisms.

Note that traffic prediction is more challenging on dataset HK-KL than datasets ST and TM; therefore, HK-KL is the default dataset for following experiments.

D. COMPARISON OF VARIANTS

To evaluate the effects of spatial attention and temporal attention in STANN, we compare it with its two variants as follows:

- TANN: There is no the spatial attention mechanism from STANN.
- SANN: There is no the temporal attention mechanism from STANN.

TABLE 6 shows the comparison in terms of RMSE, MAE and MAPE of the three models for different prediction horizons. We can see that STANN outperforms its variants in all metrics for all horizons, which indicates the important roles of the spatial and temporal attention mechanisms. Specifically, compared to SANN without temporal attention, STANN reduces RMSE and MAE at the horizon of 30-minute by 6.1% and 5.75%, respectively. Moreover, STANN records over 10% improvement on RMSE and MAE at 60, 90, and 120-minute ahead prediction, which means that its temporal attention mechanism capturing the important hidden temporal states from historical inputs improves the performance for long-term traffic prediction. On the other hand, TANN is slightly better than SANN and much worser than STANN at all horizons, since TANN ignores the spatial dependencies of network-wide links while SANN does not consider the temporal attention among historical time steps for long-term traffic prediction. Therefore, the two attention mechanisms are important for network-wide and long-term traffic prediction.

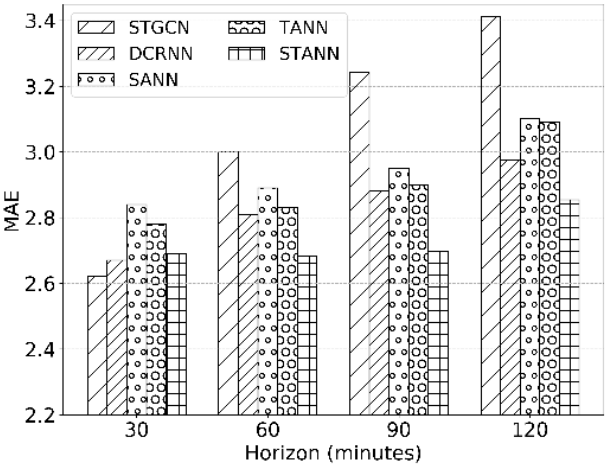

FIGURE 6. Comparisons of MAEs on dataset HK-KL during rush hours.

E. COMPUTING PERFORMANCE

We compare the training time and testing time of the proposed STANN with that of the three compared deep learning models (i.e., Seq2seq, DCRNN and STGCN) and its two variants (i.e., SANN and TANN) on the three datasets. To achieve fair comparison, we fix the same batch size as 128 for the all five models. The training time and testing time are presented in TABLE 7 and 8. Seq2Seq has the best computing performance for both training and testing since it is simply

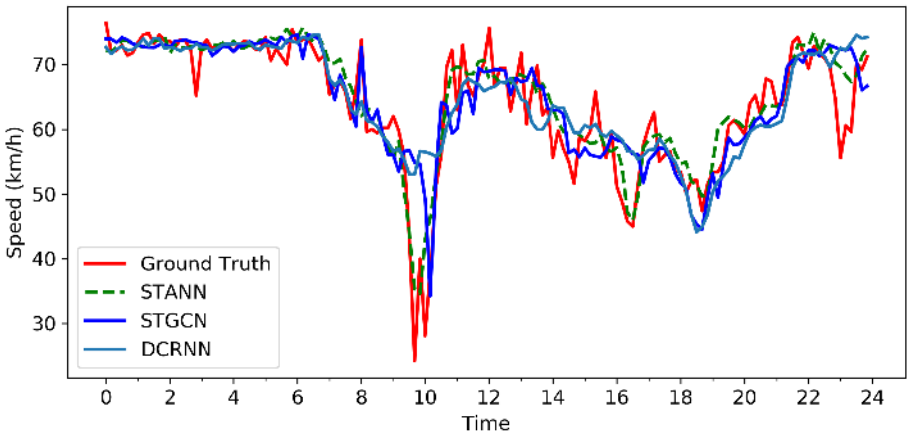

FIGURE 7. Traffic speed curves of ground truth and predictions during a whole day. (The link is located in Canal Road Flyover, Hong Kong Island).

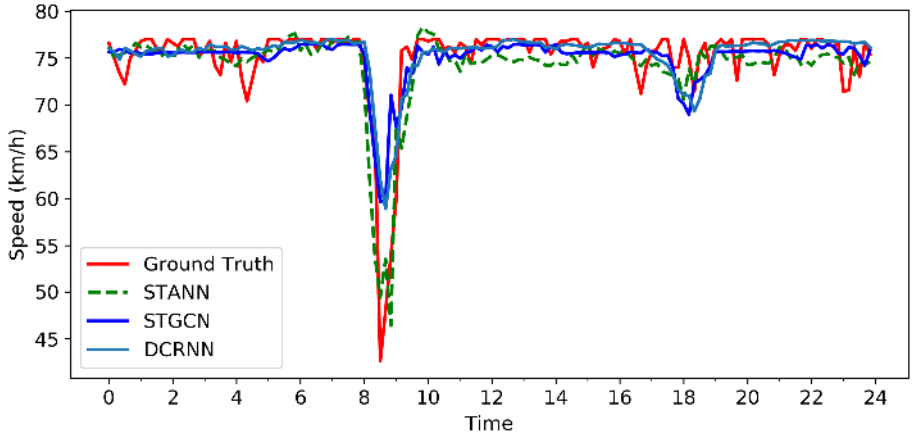

FIGURE 8. Traffic speed curves of ground truth and predictions during a whole day. (The link is located in Kai Tak Tunnel, Kowloon Peninsula).

TABLE 7. Training time consumption (minutes).

Model	HK-KL	ST	TM
SeqSeq	67	74	73
STGCN	185	179.8	188
DCRNN	1113	1076	1197
TANN	73	91	93
SANN	65	81	95
STANN	102	102	131

TABLE 8. Testing time consumption (seconds).

Model	HK-KL	ST	TM
Seq2seq	10.52	10.18	10.7
STGCN	52.67	52.38	53.95
DCRNN	184	232	261
SANN	13.38	13.58	13.62
TANN	13.97	14.02	13.95
STANN	15.57	15.54	15.56

composed of RNNs. It is clear that both the training time and testing time of DCRNN are much higher than the rest four compared models over all the three datasets. STGCN takes the second large training time and testing time among these five models for each dataset. Compared to computing performances of STGCN and DCRNN, STANN has clearly superior. On the other hand, STANN is slightly slower than its two variants and Seq2seq as it contains more information for training; the differences of testing time consumption are not clarified among these four models.

F. COMPARISON DURING RUSH HOURS

As traffic prediction during rush hours is much more challenging than other hours, we consider the performances during the morning rush hours and evening rush hours, i.e., 7am - 10am and 4pm - 7pm. Due to similar results, we only present the performance on MAE and omit that on RMSE and MAPE.

FIGURE 6 shows the comparisons of STANN with the two state-of-the-art models (i.e., DCRNN and STGCN) and its two variants (i.e., SANN and TANN) in terms of MAE for

different prediction horizons during the rush hours. From the observations in FIGURE 6, at first, it indicates that STGCN and DCRNN achieve best on MAE among all the five evaluated models at 30-minute horizon. However, they are slightly better than the proposed STANN. Then, we observe the similar result: the advantage of STANN is more significant with the increase of horizon, which further proves the superiority of STANN during rush hours. On the other hand, we can observe that the performances of its two variants TANN and SANN are better than that of STGCN whereas worser than that of STANN during rush hours at 60, 90, and 120-minute horizons; and MAE of DCRNN is slightly better than that of SANN and TANN but worser than that of STANN. Meanwhile, the margins between SANN or TANN and STANN are much clearer especially at 90 and 120-minute horizons, which further demonstrates the effectiveness of the two attention mechanisms. STGCN has the worst performances at 60, 90, and 120-minute horizons among all the evaluated models, due to its weakness of capturing temporal dependency. For example, FIGURE 7 and 8 show curves of 2-hour ahead prediction of two links on Canal Road Flyover and Kai Tak Tunnel in Hong Kong on May 8, 2018, respectively. It is obvious that the curves at non-rush hours are relatively consistent. However, during rush hours, it suggests that STANN can better catch the start point and end point of rush hours and has the smaller error than other state-of-the-art models. Therefore, the experimental results further emphasize the importance of the spatial and temporal attention mechanisms in STANN during rush hours.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed the Spatio-Temporal Attentive Neural Network (STANN) based on the encoder-decoder architecture for the network-wide and long-term traffic prediction. We first developed the spatial attention model to enhance the encoder for learning the spatio-temporal dependencies among network-wide links. Then we designed the temporal attention model to adaptively select important spatio-temporal hidden states throughout historical time steps in the decoder for multi-step and long-term traffic prediction. We evaluated STANN on the three real-world traffic datasets. Experimental results show it outperforms other state-of-the-art models, especially for long-term prediction, even during rush hours.

As we predict the network-wide network traffic, one limitation is the dimension of the spatial attention vector will be large as the network size is large. One intuitive solution is to segment the network into multiple sub-networks. We have two future research directions. First, the external factors (e.g., geographical features, point of interests, and weather) which have impacts on the traffic condition will be embedded into the new model for higher prediction accuracy. Second, we will consider the predicted traffic in real world applications. For instance, leveraging the predicted traffic condition in logistics system can reduce the cost for logistics companies.

REFERENCES

[1] E. Cascetta, *Transportation Systems Engineering: Theory and Methods*, vol. 49. Boston, MA, USA: Springer, 2013. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4757-6873-2_1

[2] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[3] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[4] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[5] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Comput. Aided Civil Infrastruct. Eng.*, vol. 14, no. 5, pp. 357–367, Sep. 1999.

[6] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[7] Y. Duan, Y. Lv, and F.-Y. Wang, "Travel time prediction with LSTM neural network," in *Proc. 19th IEEE Int. Conf. Intell. Transp. Syst.*, Nov. 2016, pp. 1053–1058.

[8] Y. Liu, Y. Wang, X. Yang, and L. Zhang, "Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, Oct. 2017, pp. 1–8.

[9] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[10] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[11] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 147–155.

[13] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[14] A. K. Ziliaskopoulos and S. T. Waller, "An Internet-based geographic information system that integrates data, models and users for transportation applications," *Transp. Res. C, Emerg. Technol.*, vol. 8, nos. 1–6, pp. 427–444, 2000.

[15] R. Fujimoto and J. Leonard, II, "Grand challenges in modeling and simulating urban transportation systems," in *Proc. Int. Conf. Grand Challenges Modeling Simulation*, 2002, pp. 1–6.

[16] A. Chow *et al.*, "TOPL: Tools for operational planning of transportation networks," in *Proc. ASME Dyn. Syst. Control Conf.*, 2008, pp. 1035–1042.

[17] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, 1984.

[18] C. Kuchipudi and S. Chien, "Development of a hybrid model for dynamic travel-time prediction," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1855, pp. 22–31, 2003. [Online]. Available: https://trrjournalonline.trb.org/doi/abs/10.3141/1855-03

[19] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Short term traffic forecasting using the local linear regression model," in *Proc. 82nd Annu. Meeting Transp. Res. Board*, 2003, pp. 1–30.

[20] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 253–261, 2001.

[21] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 3, pp. 211–234, 2005.

[22] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *J. Transp. Eng.*, vol. 132, pp. 114–121, Sep. 2006.

[23] J. Van Lint, S. Hoogendoorn, and H. Van Zuylen, "Freeway travel time prediction with state-space neural networks: Modeling state-space dynamics with recurrent neural networks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1811, pp. 30–39, Jan. 2002.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.
[26] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vol. 18, 2018, pp. 984–992.
[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[28] B. Liao *et al.*, "Deep sequence learning with auxiliary information for traffic prediction," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 537–546.
[29] B. Liao, S. Tang, S. Yang, W. Zhu, and F. Wu, "Multi-modal sequence to sequence learning with content attention for hotspot traffic speed prediction," in *Proc. Adv. Multimedia Inf. Process., Pacific-Rim Conf. Multimedia*, 2018, pp. 212–222.
[30] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[31] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, 2000.
[32] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[33] W. H. K. Lam, K. S. Chan, and J. W. Z. Shi, "A traffic flow simulator for short-term travel time forecasting," *J. Adv. Transp.*, vol. 36, no. 3, pp. 265–291, 2002.
[34] M. L. Tam and W. H. K. Lam, "Using automatic vehicle idenification data for travel time estimation in Hong Kong," *Transportmetrica*, vol. 4, no. 3, pp. 179–194, 2008.

ZHIXIANG HE received the M.S. degree from Shenzhen University, Shenzhen, China, in 2017. She is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. Her research interests include traffic data mining, intelligent transportation systems, and recommender systems.

CHI-YIN CHOW received the M.S. and Ph.D. degrees from the University of Minnesota-Twin Cities, USA, in 2008 and 2010, respectively. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong. His research interests include big data analytics, data management, GIS, mobile computing, location-based services, and data privacy. He received the VLDB "10-year award," in 2016, and the best paper awards in ICA3PP 2015 and IEEE MDM 2009. He is the Co-Founder of ACM SIGSPATIAL MobiGIS, and was the Co-Chair of the ACM SIGSPATIAL MobiGIS, from 2012 to 2016, and the Editor of the ACM SIGSPATIAL Newsletter.

JIA-DONG ZHANG received the M.Sc. degree from Yunnan University, China, in 2009, and the Ph.D. degree from the City University of Hong Kong, in 2015. He is currently a Research Fellow with the Department of Computer Science, City University of Hong Kong. His research work has been published in premier conferences (e.g., ACM SIGIR, CIKM, and SIGSPATIAL), transactions (e.g., ACM TIST, IEEE TKDE, TDSC, TSC, and TITS), and journals (e.g., the IEEE Access, *Pattern Recognition*, and *Information Sciences*). His research interests include deep learning, data mining, recommender systems, and location-based services.