24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Deep Stacked Residual Neural Network and Bidirectional LSTM for Speed Prediction on Real-life Traffic Data

Sayda Elmi [1]

Abstract. Forecasting accurate traffic speed is of great importance to advanced traffic management systems but challenging problem as it is affected by many complex factors, such as events, inter-road traffic, traffic lights, period and weather conditions. Traffic speed prediction based on deep learning techniques has received much attention in recent years. However, the power of deep learning methods has not yet fully been exploited in traffic prediction in terms of the depth of the model architecture. This paper designs a deep-learning-based structure, called BiRNet, to forecast accurate traffic speed in each and every region in a city. More specifically, we employ the residual neural network and the bi-directional recurrent neural network to model the spatial and temporal closeness, respectively. A look-up layer is introduced to model the spatial scale of the prediction area. BiRNet learns to dynamically aggregate the output of these neural networks which is further combined with external factor learning to improve the spatially correlated time series data forecasting. Our extensive experiments on real-world trip data-sets generated in NYC's road network, show that our proposed deep learning algorithm significantly outperforms the state-of-the-art learning algorithms.

1 Introduction

Fortunately, the positioning and wireless technologies have resulted in significant amount of trajectory data in diverse range of domains [31] which boosts traffic analysis such as travel time estimation. Although all electronic maps and online car-hailing services provide the travel time estimation such as Google Map, Uber and Didi, the estimation quality is critical to the user experience of these applications. Traffic speed Prediction (TSP) is a crucial task to get accurate travel time estimation. Therefore, the problem of TSP has attracted many researcher's attention and been widely studied in intelligent transportation systems but providing an accurate speed prediction is still very challenging.

In this paper, we aim to predict the future speed of each road segment in a large road network based on historical observations. For this end, we propose a deep spatio-temporal and contextual neural network called BiRNet, to provide accurate traffic speed prediction.

For each road segment, the fluctuation of its speed usually follows some patterns.

Spatial Patterns: Different road segments have strong dependency with each other. For example, a congestion in a given road will affect surrounding areas especially adjacent roads. Likewise, speed on a road tends to slow down if congestion occurs in its surrounding area. Therefore, the traffic speed prediction model must deal with spatial correlations among different road segments. For this reason, some recent studies adopted the convolutional neural network (CNN) to learn traffic as images [15, 29]. They have achieved significant success since CNN has shown a powerful ability to model similarity between pixels [11], as a kind of spatial relations. However, this is insufficient for TSP problem, since there are more patterns that should be learned.

Temporal Patterns: The temporal information is also a key factor that has a great impact on traffic speed prediction. For example, the speed will be much more higher in the rush hours than the off-peak hours as well as in the weekend. Thus, the traffic speed is highly correlated with the time period in a year, a month and a day, rush hour or holiday. These kind of patterns can be discovered by existing time-series models, such as hybrid auto-regressive integrated moving average (H-ARIMA) [17] and support vector regression (SVR) [22]. Recently, recurrent neural network (RNN) has demonstrated a good performance in both stability and accuracy in terms of traffic speed prediction [19, 16]. However, these models, which usually treat traffic speeds as sequential data, ignore the spatial feature of transportation networks.

Context Patterns: Some external factors, such as driver profile, traffic lights, events and weather conditions may effect the traffic speed prediction.

To tackle these challenges, our proposed spatio-temporal model, BiRNet, fully investigates the potential of deep learning techniques in terms of depth by jointly training residual neural networks and recurrent neural networks together to combine their benefits and collectively predict traffic speed in every region in a city. It balances the memorization and representation abilities in one model and effectively alleviates the limitations in existing methods. Our contributions can be summarized as follows:

1. To learn the spatial traffic dynamics, BiRNet uses convolution-based residual networks since it can effectively capture nearby and distant spatial dependencies between roads in a city region. To model the structure of the road network, we construct an adjacent road matrix. A look-up operation is used to embed the road network structure in the model. This operation is used to learn meaningful features from adjacent road matrix.
2. Thanks to its great success in sequence learning tasks, Recurrent Neural Network (RNN) is adopted by BiRNet. More specifically, a Bidirectional Long Short-Term Memory (Bi-LSTM) is used to learn time-series patterns.
3. To provide more accurate prediction, BiRNet combines residual neural networks and Bi-LSTM in a rational way to make use of their advantages. In addition, BiRNet dynamically aggregates the output of the aforementioned networks with external factors learning.
4. We conduct extensive experiments on real-world large-scale trip

[1] National University of Singapore, School of Computing, Singapore. saida@comp.nus.edu.sg

data-sets on NYC's road network generated by thousands of main links. The proposed approach clearly outperforms state-of-the-art methods for traffic speed prediction.

2 Related Work

In recent years, number of intelligent traffic speed prediction systems have been proposed where the algorithms were relied on statistics theories. Authors in [18, 10] proposed a spot speed measurement methods based on loop detectors to measure traffic stream speeds over road segments at fixed locations along a road. However, the approach proposed in [20] falls into the line of route-based method facing the route mapping challenge as well as the data sparsity problem. Regression models are the most commonly used method to estimate traffic and are based on regression analysis. A support vector regression (SVR) based-method is used in [1, 23] to perform large scale traffic prediction. Authors in [12] conducted analysis on Taxi trajectories and used Random Forest (RF) based-approach to improve the traffic prediction.

Nevertheless, these techniques were unsatisfying to predict the accurate traffic speed since they ignore the important spatio-temporal and contextual features of transportation networks. Due to the large number of factors having direct impact on the traffic delay, there are no accurate mathematical models describing the relationship between the traffic speed and its influencing factors. Therefore, the TSP becomes a complex problem due to the large number of factors that could affect traffic dynamics. For this reason, many researchers have recently been attracted towards machine learning and more specifically deep learning methods.

Deep learning techniques have been successfully applied to various applications [27, 19] to determine complex non-linear relationships between variables. Specially, artificial Neural Networks (NN) have been widely used in traffic engineering. Authors in [26] developed a back propagation neural network (BPNN)-based approach to predict trip-oriented travel time for Origin-Destination pairs in urban network. As one of famous deep learning techniques, RNN achieves great success in sequence learning tasks [24]. In [8], the author introduced LSTM which enables RNN to learn long-term sequence information. Interest in LSTM has greatly increased in recent years and have been successfully applied to solve traffic problems [21, 6, 4] since it can automatically reserve historical sequence information in its model structure. Authors in [28] introduced a bidirectional LSTM-based approach by utilizing backward information to improve the memory capability. These methods achieved better results than statistical methods. However, these attempts still mainly focus on the temporal correlations of traffic evolution at a single location. Therefore, convolutional neural networks have been successfully applied in TSP [30, 15] considering the spatial correlations from the road network. To have a very deep structure modeling the spatial features, authors in [29, 13], proposed the residual learning (ResNet). Authors in [14] proposed a spatio-temporal model using both convolutional and recurrent neural networks.

Compared to existing TSP models, our hybrid architecture fully investigates the potential of deep learning techniques in terms of depth by taking advantages of both residual component and bi-LSTM blocks. This combination is exploited in a rational way to learn spatial and temporal patterns, respectively and then improve accuracy.

3 Methodology

In this section, the components and the architecture of the proposed BiRNet are described in details. We first define the TSP problem and give out the formulation of network-based traffic speed prediction.

3.1 Network-wide Traffic Speed Data

A vehicle speed and position at a certain time can be provided by a vehicle trajectory recorded by a dedicated GPS device car. For each road segment, we compute the average traffic speed from the GPS pings as shown in Figure 1.

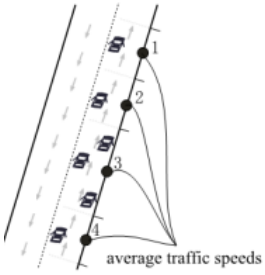


Figure 1: Average Traffic Speeds

The input of TSP problem at one location can be represented by the vector X_t which is a sequence of speed values x with n historical time steps where

$$X_t = [x_{t-n}, x_{t-(n-1)}, \ldots, x_{t-2}, x_{t-1}, x_t] \tag{1}$$

But, the traffic speed at one location can be impacted by nearby and faraway locations. For this reason, we design a spatio-temporal matrix as the input of our model. Traffic speed of road r at time t can be defined as $x_{r,t}$. Let $|\mathcal{R}|$ be the number of roads in a given network $\mathcal{R}$, the matrix can be expressed as follows:

$$X_{r,t} = \begin{bmatrix} x_{r_1,t-n} & \cdots & x_{r_1,t-1} & x_{r_1,t} \\ x_{r_2,t-n} & \cdots & x_{r_2,t-1} & x_{r_2,t} \\ \vdots & \ddots & \vdots & \vdots \\ x_{r_{|\mathcal{R}|-1},t-n} & \cdots & x_{r_{|\mathcal{R}|-1},t-1} & x_{r_{|\mathcal{R}|-1},t} \\ x_{r_{|\mathcal{R}|},t-n} & \cdots & x_{r_{|\mathcal{R}|},t-1} & x_{r_{|\mathcal{R}|},t} \end{bmatrix} \tag{2}$$

As in equation 2, the row vector contains traffic speed in the same road from continuous time intervals (from t to t-n), the columns contain traffic speed in all $|\mathcal{R}|$ roads at the same time interval. Note that $X_{r,t}$ is of dimension $|\mathcal{R}| \times n$.

In this paper, we would like to predict the estimated traffic speed of all roads in a network at a future time. More formally, given $X_{r,t}$ and z where z is the number of intervals, we want to predict $X_{r,t+i}$ for $1 \leq i \leq z$.

3.2 Model Architecture

As shown in Figure 2, the architecture of BiRNet is comprised of four major components which are: (i) several residual layers to model the spatial evolution, (ii) a recurrent layer and more specifically, we use the Bi-LSTM component to model the temporal dependency, (iii) a periodicity extraction layer and (iiii) a context extraction layer modeling the external influence such as weather and holidays..

For learning spatial patterns, BiRNet employs Residual units since it has very deep structure that can effectively capture the spatial traffic dynamics from nearby and distant regions. As an input, residual layers take the traffic speed matrix $X_{r,t}$. This component shares the same road network topology with a convolutinal neural network in each residual unit. Such structure captures the spatial traffic dynamics of the surrounding areas. After extracting the spatial patterns, we reshape these features to be suitable for time-series learning and feed into RNN layer and more specifically, a Bi-LSTM as a deeper variant of RNN for sequence modeling.

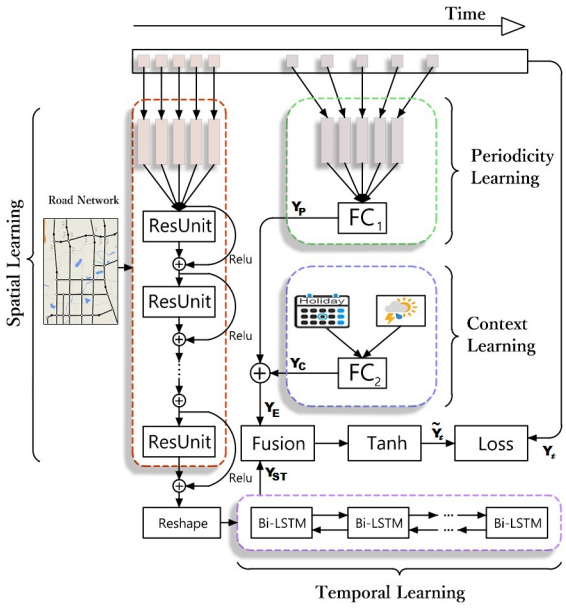


Figure 2: BiRNet: A Hybrid Architecture.

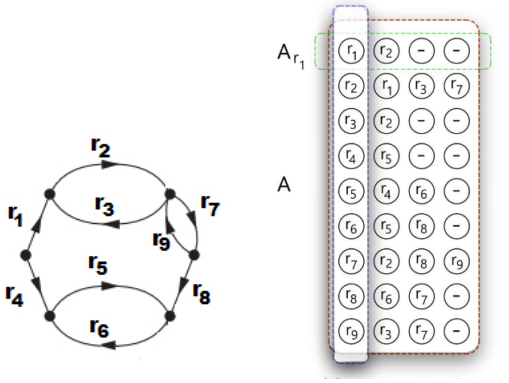
In addition, the traffic speed prediction is directly affected by periodicity features. For instance, a traffic congestion occurring at 8am will be almost the same as tomorrow at 8am. It means that traffic conditions during morning rush hours may be similar on consecutive workdays/weeks, repeating every 24 hours. Hence, we feed the speed vectors of periodic time intervals into fully-connected layers (FC) to capture such temporal dynamics.

For the external learning, we extract context factors such as weather, event, holiday and so on from external data-sets, feeding them into FC layers. By a Tanh function, the outputs are aggregated.

3.3 Spatial Features Extraction

Thanks to its good performance on capturing spatial features from adjacent pixels, convolution neural network has been widely used in image data analysis [11]. Since traffic evolution is restricted by the road network, i.e., the speed of a given road is impacted by the traffic in adjacent road segments, we thus design a convolution layer that embed the topology of road network into convolution to capture more meaningful spatial features. To have a super deep convolution structure of 100 layers, even over-1000 layers, we use the deep residual learning [9] which has shown state-of-the-art results on multiple challenging recognition tasks, including image classification, localization, segmentation and object detection[9].

The Road Network as Graph: Let $\mathcal{R}$ be the set of road segments



(a) Road Network (b) Adjacent Road Matrix

Figure 3: Road Network and Adjacent Road Matrix.

r. We use $r.s$ and $r.e$ to represent the start and end of the road $r \in \mathcal{R}$. To represent the road network topology (Figure 3a), we use an adjacent road matrix as shown in Figure 3b, denoted by A, which records all adjacent roads for each road $r \in \mathcal{R}$ by A_r where

$$A_r = \{r, r' \in \mathcal{R} | r.s = r'.e \; or \; r.e = r'.s\} \tag{3}$$

Then A contains all $A_r | r \in \mathcal{R}$ and has the dimension of $|\mathcal{R}| \times D$, where $D = max\{|A_r|, \forall r \in \mathcal{R}\}$, i.e., the maximum number of adjacent roads for all roads in $\mathcal{R}$. For the road network represented in Figure 3, we have $|\mathcal{R}| = 9$ and $D = 4$ since we consider the directly connected roads to r and also r itself. Since the adjacent segment sets for different roads have different sizes, we pad each empty column in A with a virtual road which has no adjacent roads and has a very small speed value so that when it is scaled, its value becomes -1. During training and testing we are including virtual road in the input training data, but on the output of training data, we remove this virtual road, so our network is not learning to predict virtual road but it is acting as padding in our data-set.

Residual Unit: The components of the residual unit are given in the Figure 4. This later contains three combinations of "LK + Conv + BN". First, we feed the speed matrix $X_{r,t}$ into the residual block. A look-up layer (LK) is required to embed the road network topology. A stack of convolutions is used in a single residual unit to understand connections between locations with a far distance. We attempt Batch Normalization (BN) after Conv layer for faster training speed.

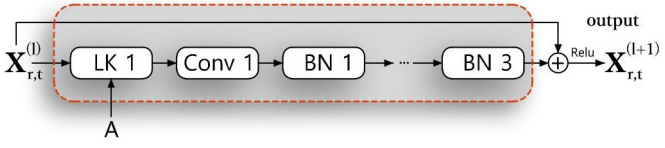


Figure 4: Residual Unit.

Look-up Operation: Given the traffic speed data recorded in $X_{r,t}$, the topology of road network represented by the adjacent road matrix A is embedded as follows:

$$\mathcal{X}^{(l)} = LK(X_{r,t}^{(l)}, A) \tag{4}$$

This operation [25] aims to perform a look-up in the embedding matrix and return the embedding of A. An embedding matrix would look like a vector of speed values of the adjacent roads in A from

the speed matrix $X_{r,t}$. $\mathcal{X}^{(l)}$ represents the result of the look-up operation and contains not only the speed values of $|\mathcal{R}|$ roads over n time-slots, but also the speed values of their adjacent roads and has the dimension $|\mathcal{R}| \times D \times n$. Note that the virtual added road is acting as a placeholder in LK operation. Figure 5 describes the LK result

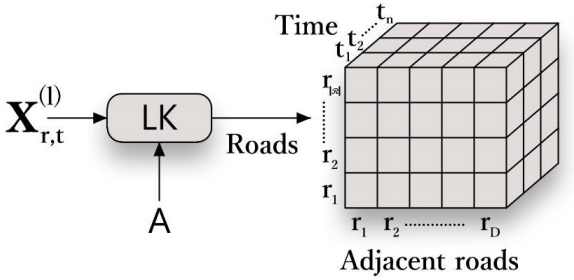


Figure 5: Look-Up Operation.

with more details. To model traffic conditions, LK constructs a tensor $\mathcal{X}^{(l)} \in \mathbb{R}^{|\mathcal{R}| \times D \times n}$, with the three dimensions standing for road segments, adjacent roads and time-slots, respectively. As shown in Figure 5, an entry $\mathcal{X}^{(l)}(i,j,k) = x$ denotes the speed value of the j^{th} adjacent road of the road i in time-slot k.

Convolution: The input of the convolution layer is traffic road tensor $\mathcal{X}^{(l)}$. We use a 3D convolution layer to extract the features of Spatial matrix as shown in Figure 6. Output layer is to generate the prediction result. At an arbitrary l-th layer, we use $k^{(l)}$ filters to convolve and concatenate all matrices to get $\mathcal{X}^{(l+1)}$. The k-th matrix convolved by the k-th filter can be formulated as follows

$$\mathcal{X}^{l,k} = [x_1^{l,k}, \ldots, x_i^{l,k}, \ldots, x_{|\mathcal{R}|}^{l,k}] \tag{5}$$

where

$$x_i^{l,k} = f\big(LK(X_{r,t}^{(l-1)}, A) * W^{l,k} + b^{l,k}\big) \tag{6}$$

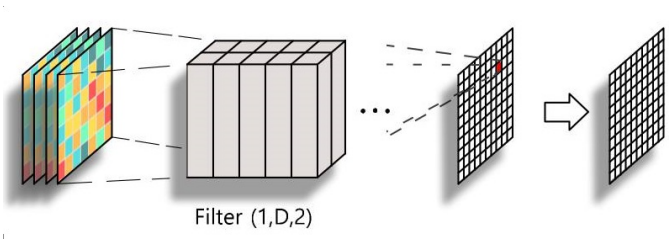


Figure 6: 3D Convolution.

Here, $*$ denotes the convolution operation which uses the k-th filter $W^{l,k}$, f is an activation function, e.g. the rectifier ReLU $f(x) = max(0,x)$ which has achieved a training effectiveness in reducing the problem of gradient vanishing, $W^{l,k}$ and $b^{l,k}$ are the learnable parameters in the l-th layer with the k-th filter. We use a 3D filter of dimension $1 \times D \times 2$ where D is the columns number of adjacent road matrix A. This filter aims at emphasizing the spatial features of adjacent roads.

To model large citywide dependencies and connect far roads, we still need deep networks to have more consecutive layers. To address this issue, we employ residual learning [9] in our model. We stack several residual units as follows:

$$X_{r,t}^{(l+1)} = X_{r,t}^{(l)} + \mathcal{F}\big(X_{r,t}^{(l)}, \theta^{(l)}\big) \tag{7}$$

where $X_{r,t}^{(l)}$ and $X_{r,t}^{(l+1)}$ are the input and output of the l-th residual unit, respectively; $\mathcal{F}$ is the residual function, e.g. a stack of three combinations of "LK + Conv + BN" as shown in Figure 4 and $\theta^{(l)}$ includes all learnable parameters in the l-th residual unit.

3.4 Temporal Features Extraction

After K residual units, we use the RNN model to learn the long-term temporal patterns considering the surrounding area. On top of the K residual units, we get the output tensor $X^K \in \mathbb{R}^{p \times |\mathcal{R}| \times f_K}$ where f_K is the number of the convolution filters at the last K-th Conv layer and p is the period (the number of days/weeks). We reshape X^K in the way of time sequence to feed into RNN layer. We get a tensor $X' \in \mathbb{R}^{|\mathcal{R}| \times n \times f_K}$ representing the road speed vectors for all n time-slots, where

$$X'_{i,t} = X^K[t,i,:] \tag{8}$$

As a variant of RNN, LSTM is good at learning long time-series patterns avoiding the problem of vanishing gradient [2]. Compared with LSTM, bidirectional LSTM (Bi-LSTM) [7] utilizes additional backward information and thus enhances the memory capability. We follow the same way to capture the temporal information of every time interval.

Traffic speed prediction is affected by daily, weekly, or occasional events. For instance, traffic patterns on Monday and Friday may be very different though both are workdays. Thus, to learn the long and short term traffic dynamics, we train the speed vector $X'_{i,t}$ which records the speed on road $r_i \in \mathcal{R}$ for n time-slots.

The structure of a bidirectional LSTM can be represented by two type of connections, one going forward in time, which helps us learn from previous representations and another going backwards in time, which helps us learn from future representations.

When feeding $X'_{i,t}$ into Bi-LSTM, we get the i-th hidden states $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ of the forward and backward layers, respectively. We then concatenate these two states to get the i-th hidden state $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$.

For a given LSTM cell, the input gate, the forget gate, the output gate and the input cell state can be calculated using the following equations:

$$\begin{aligned} i_t &= \sigma_g\big(w_i X'_{i,t} + u_i h_{t-1} + b_i\big) \\ f_t &= \sigma_g\big(w_f X'_{i,t} + u_f h_{t-1} + b_f\big) \\ o_t &= \sigma_g\big(w_o X'_{i,t} + u_o h_{t-1} + b_o\big) \\ g_t &= tanh\big(w_g X'_{i,t} + u_g h_{t-1} + b_g\big) \end{aligned} \tag{9}$$

where w_i, w_f, w_o and w_g the weight matrices mapping the hidden layer input to the three gates and the input cell state, while u_i, u_f, u_o and u_g are the weight matrices connecting the previous cell output state to the three gates and the input cell state. b_i, b_f, b_o and b_g are four bias vectors. The σ_g is the gate activation function, e.g. the $Sigmoid$ function $\sigma(a) = 1/(1+e^{-a})$, and the $tanh$ is the hyperbolic tangent function. Then the memory cell and hidden state are updated as:

$$\begin{aligned} c_t &= f_t * c_{t-1} + i_t * g_t \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{10}$$

The forward layer output sequence, $\overrightarrow{h}_i$, is iteratively calculated using inputs in a positive sequence from time $t-n$ to time t, while the backward layer output sequence, $\overleftarrow{h}_i$, is calculated using the reversed inputs from time $t-n$ to time t. The Bi-LSTM layer generates an

output vector, Y_T in which each element is calculated by using the following equation:

$$Y_i = \sigma(\overrightarrow{h}_i, \overleftarrow{h}_i) \tag{11}$$

where σ function is used to combine the two output sequences. Similar to the LSTM layer, the final output of a Bi-LSTM layer can be represented by the Spatio-Temporal vector Y_{ST} in which the last element is the predicted speed for the next time iteration such as

$$Y_{ST} = [Y_{i,t-n}, \ldots, Y_{i,t}, Y_{i,t+1}] \tag{12}$$

3.5 Context and Periodicity Learning

In addition to the spatial and temporal factors, the traffic speed prediction is affected by many complex factors such as weather conditions, holidays and Periodicity. We extract and learn these information to have more accurate prediction.

Context Extraction Layer: Observations on traffic dynamics have demonstrated that the traffic speed is significantly influenced by both weather conditions and holidays [14, 29]. This is because the traffic speed in the heavy rain will be slower than that in sunny days because of safe driving. On the other hand, the traffic speed during holidays can be different from during normal days. In our experiment, the holidays can be directly extracted from our data-set, however, weather data are extracted from public data-sets using a parser. We incorporate the attributes of the weather conditions (rainy/sunny/windy etc.), holidays and the time information (day of the week). Note that these factors are categorical values which can not feed to the neural network directly. In our model, we use one hot encoding method to convert categorical data to a low-dimensional integer vector. Formally, a fully connected layer is adopted to extract context information and get Y_C.

Periodicity Extraction Layer: If we consider a single road segment, we can observe that the traffic dynamic changes periodically. We mainly consider three kinds of periodicity:

- Inter-time-intervals Periodicity: The traffic condition in a region is affected by recent time intervals, both near and far. For instance, a traffic congestion occurring at 8am will affect that of 9am. We construct X_i to describe the time-intervals dependency.
- Daily Periodicity: we can see the daily periodicity denoted by X_d, especially at weekdays, i.e., the traffic average speed at a certain time interval (17:00pm-17:15pm) of Wednesday can be similar to the next following day, in the absence of special condition.
- Weekly Periodicity: at a certain time interval (8:00pm-8:30pm) of Monday, the traffic speed shows a similar trend at the same time interval of the previous Monday on previous weeks, which is the weekly periodicity denoted by X_w.

Using a fully connected layer, we train X_i, X_w and X_d to get Y_i, Y_w and Y_d, respectively. The speed prediction respecting the periodicity factor Y_P is obtained such as $Y_P = Y_w \oplus Y_d \oplus Y_i$ where $\oplus$ is the concatenation of the results to get Y_P. Then, as shown in Figure 2, we integrate it with Y_C from the context learning to get the external vector Y_E that has the same shape of Y_{ST}.

3.6 Fusion

Using the spatio-temporal learning is not enough to get more accurate speed prediction. For this reason, we merge the output of the first two components, i.e., stacked residual neural networks and bidirectional LSTM, with that of the external component as shown in Figure 2. Finally, the predicted value at the t-th time interval, denoted by $\widetilde{Y}_t$, is defined as

$$\widetilde{Y}_t = tanh(W_{ST} \circ Y_{ST} + W_E \circ Y_E) \tag{13}$$

where $\circ$ is Hadamard product (i.e., element-wise multiplication), W_{ST} and W_E are the learnable parameters.

To predict Y_t, our model can be trained by minimizing mean squared error $\mathcal{L}(\theta)$ between the predicted speed matrix and the true speed vectors such as

$$\mathcal{L}(\theta) = ||Y_t - \widetilde{Y}_t||_2^2 \tag{14}$$

where θ are all learnable parameters in BiRNet model.

4 Experiment

To evaluate the performance of our deep learning model BiRNet, we conduct a series of experiments on road networks in New York City. In this section, we first discuss the setup of our experiments, then we present the baseline methods and finally, we discuss the experimental results.

4.1 Data Description

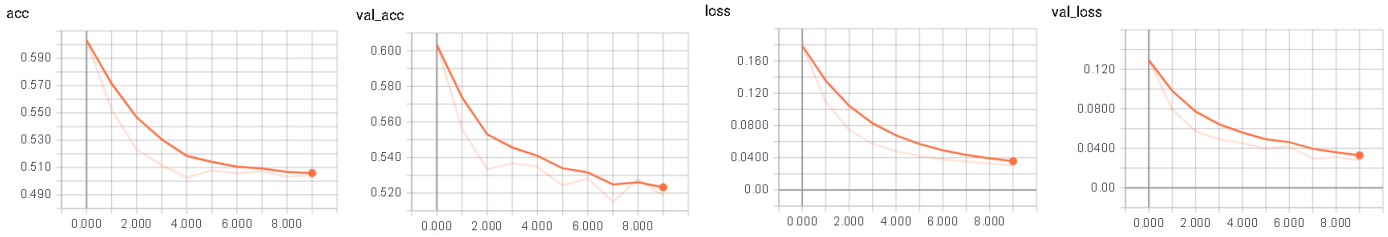
In our experiment, we used the following data-sets to test our BiRNet model. Details are given as follows:

- UIUC New York City Traffic Estimates[2]: a public data-set, on which we extract spatial features and conduct our following experiments. This data-set covers 700 million taxi trips from 2010 to 2013 in New York City. It contains accurate hourly average traffic speed measurement for almost all road segments of the NYC road networks. The road network includes about 261 thousand main links and 96 thousand nodes. Only holidays information are used for the context learning. Among the data of the year 2013, three months data are used for evaluating the model, the last three weeks are test set.

Preprocessing: We represent the road network as a directed graph composed of road segments and nodes, each node is an intersection of the road network. Each directed link is a road segment connecting two nodes. A road/street consists of multiple links, two-way streets are often represented as two directed links with opposite directions. We calculate the average traffic speed for each link at a particular hour. To learn the short-term traffic periodicity, we divide a day δ into different time-bins (e.g. an hour is a bin in our experiments). An hour is also divided by 4 time-slots τ, (e.g., each is 15 minutes).

To scale the data into the range $[-1, 1]$, we use the Min-Max normalization method and then, for comparison with real speed data, we re-scale the predicted value back to the normal values. For external factors, a binary vector is given by one-hot coding to transform and represent the holidays, the weather conditions and the time-stamp of the day. We also scale the Temperature and Wind speed into the range $[0, 1]$ by Min-Max normalization. We use tanh in the output of the BiRNet model as our final activation (see Equation 13). We train our network with the following hyper-parameters setting: mini-batch size (48), learning rate (0.0001) with adam optimizer, $1 \times$ D filters (32) and $2 \times$ D filters (16) in each Convolution layer. Afterwards, we continue to train the model on the full training data for a fixed number of epochs (e.g., 100, 500, 1000 epochs). To learn the periodicity dynamics, we empirically fix it to one-day and one-week.

[2] https://lab-work.github.io/data/



(a) Training Accuracy (b) Validation Accuracy (c) Training Loss (d) Validation Loss

Figure 7: Training Statistics.

4.2 Benchmarks

Several prevailing algorithms are chosen for comparisons with our proposed model BiRNet.

- **SVR** [22]: Support Vector Regression (SVR) is a powerful regression method.
- **ARIMA** [17]: Auto-Regressive Integrated Moving Average (ARIMA) is a well-known model for predicting future values in a time series.
- **RNN** [24]: A recurrent neural network (RNN) is a method to a temporal sequence values.
- **GRU** [5]: Gated Recurrent Unit (GRU) is an improved version of standard recurrent neural network to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.
- **LSTM** [16]: Long short-term memory (LSTM) were developed to deal with the vanishing gradient problems that can be encountered when training traditional RNNs.
- **Bi-LSTM** [3]: Using bidirectional LSTM will run the inputs in two ways, one from past to future and one from future to past to measure the backward dependency of traffic data prediction.
- **DCNN** [15]: uses convolution, pooling and fully connected layers to predict traffic speed.
- **ST-ResNet** [29]: uses residual network to model three temporal properties for traffic prediction.
- **LC-RNN** [14]: uses convolution, RNN and fully connected layers for speed prediction.

4.3 Performance

Evaluation Metrics:

We measure our method by Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as follows:

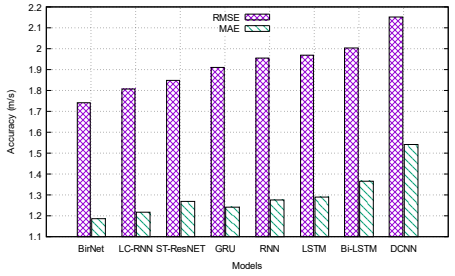
$$RMSE = \sqrt{\frac{1}{n}\Sigma_{t=1}^{n}\left(y_t - \hat{y}_t\right)^2} \tag{15}$$

$$MAE = \frac{1}{n}\Sigma_{t=1}^{n}|y_t - \hat{y}_t| \tag{16}$$

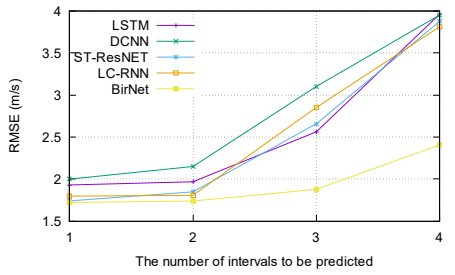
$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{17}$$

where $\hat{y}_t$ and y_t are the predicted value and real value, respectively; n is the number of all predicted values.

Results on New York City Data-set: The Figure 8a shows the RMSE and MAE of the most existing models solving the traffic speed prediction problem. Our proposed model BiRNet outperforms the existing approaches.



(a) Prediction performance ($p = 2$)

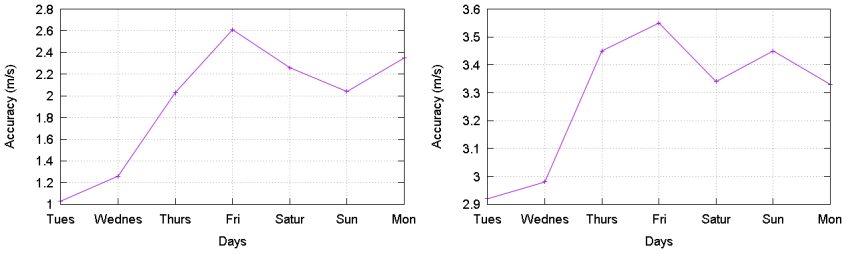


(b) Predicted intervals effect

Figure 8: Results on NYC Data-set.

More methods are shown in Table 1. First we compare our model with 9 other benchmarks using NYC data-set. The time interval is 15 min and the number of intervals to be predicted is 2. We observe that our proposed BiRNet model is better than 9 benchmarks, it outperforms nine well-known existing methods. In addition, to analyze the effect of the different components in our model, we propose the structures BiRNet$\backslash P$, BiRNet$\backslash C$ and BiRNet$\backslash LK$ that remove the components: periodicity learning, context learning and spatial learning, respectively. Results of BiRNet$\backslash LK$ shows that discarding look-up layer may reduce the accuracy achieved by BiRNet. That means, the topology of road network captured by our look-up layer improves the results and demonstrates the effectiveness of taking spatial dependency into consideration. In addition, comparing BiRNet to BiRN$et \setminus P$ and BiRNet$\backslash C$, we can observe that the result is further promoted using periodicity and context learning components, respectively, pointing out that external extraction layer is beneficial for speed prediction.

The default parameter of the number of predicted intervals, denoted p, is 2. Figure 8b shows the comparative performances for LSTM, DCNN, ST-ResNet and BiRNet varying the number of predicted intervals from 1 to 4. As shown in the figure, BiRNet achieves the best performance when varying p. Note that the performance becomes worse with larger p since the correlation between speed



(a) Residential Road (b) Trunk Road

(c) Motorway (d) Motorway

Figure 9: Deep Analysis on different road types.

[ht]

Methods	RMSE	MAE	MAPE
SVR	3.452	3.132	0.367
ARIMA	4.763	4.612	0.385
RNN	1.954	1.277	0.128
GRU	1.909	1.243	0.132
LSTM	1.969	1.291	0.173
Bi-LSTM	2.180	1.367	0.136
DCNN	2.151	1.541	0.158
ST-ResNEt	1.849	1.269	0.131
LC-RNN	1.808	1.216	0.125
BiRNet [ours]	**1.740**	**1.186**	**0.122**
BiRNet$\backslash P$	1.797	1.231	0.124
BiRNet$\backslash C$	1.781	1.222	0.124
BiRNet$\backslash LK$	1.756	1.205	0.122

Table 1: Results on NYC Data-set.

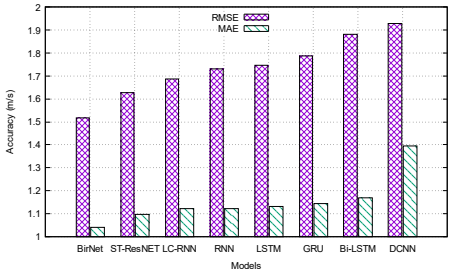
in time intervals being predicted and speed in current moment decreases.

Deep analysis:

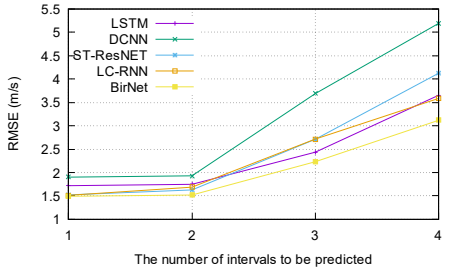
In the following, we first describe some training statistics and then present our model performance for different type of roads.

In Figure 7, we can see how our model did over time (10 epochs). First, we show accuracy behaviour on training and validation set in Figures 7a and 7b, respectively. Interestingly enough, our validation accuracy still continued to hold. Figures 7c and 7d show the shape of training and validation loss, respectively. Loss is the measure of error, the lower the loss, the better a model. It clearly looks that validation loss is improving over epochs.

We analyze the traffic pattern at different scenarios. Figure 9 shows the model performance on different road types (motorways, residential ways and trunk ways) on different days (weekday and weekend). This is a traffic speed prediction at 8:15 am on testing set from 25-th December to 31-st December 2018. The days in the figures are in chronological order, i.e., 25-th December 2018 is Tuesday, etc. The accuracy of our model over different road type is the difference between predicted speed values and real values. We use



(a) Prediction performance ($p = 2$)



(b) Predicted intervals effect

Figure 10: Results on NYC Data-set.

10 epochs for training the model. Through Figure 9, we can observe the performance of our model on residential road, trunk road and motorways. On different road type, the BiRNet model has a good performance on Tuesday, 25-th December 2018, which shows our model capacity to learn special pattern as holidays. For residential and trunk roads, the accuracy value slightly increases over week days but decreases on Saturday and Sunday which shows that our model is good at learning temporal patterns in both residential and trunk roads. For motorways, our model has shown a similar behaviour and the accuracy values slightly increases and decreases over days.

In summary, from the above experimental result on NYC data set, we can observe that BiRNet achieves the best performance compared with the state-of-the-arts. Moreover, the fusion of spatial dependency on the road network, temporal evolution and contextual information are the key factor to more learn the traffic speed behaviour.

5 Conclusion

This paper proposes a novel deep learning based traffic speed prediction method that can extract spatio-temporal traffic features using a look-up layer to embed the road network and learn surrounding area dynamics. The model also integrates external information such as periodicity, holidays and weather data. We evaluate our model on two road networks in NYC, achieving performances which are significantly beyond nine existing methods, confirming that our model is applicable to the traffic speed prediction problem.

REFERENCES

[1] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fathi, M. Xu, M. M. Dhanya, N. Mitrovic, and P. Jaillet, 'Spatio-temporal patterns in large-scale traffic speed prediction', *IEEE Transactions on Intelligent Transportation Systems*, **15**(2), (2014).

[2] Y. Bengio, P. Simard, and P. Frasconi, 'Learning long-term dependencies with gradient descent is difficult', *IEEE Transactions on Neural Networks*, **5**(2), 157–166, (1994).

[3] Zhiyong Cui, Ruimin Ke, and Yinhai Wang, 'Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction', *CoRR*, **abs/1801.02143**, (2018).

[4] Yanjie Duan, Yisheng Lv, and Fei-Yue Wang, 'Travel time prediction with lstm neural network', in *Proceeding of the 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1053–1058, (2016).

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

[6] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang, 'Identifying human mobility via trajectory embeddings', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 1689–1695, (2017).

[7] A. Graves and J. Schmidhuber, 'Framewise phoneme classification with bidirectional lstm networks', in *Proceedingsof IEEE International Joint Conference on Neural Networks.*, volume 4, pp. 2047–2052 vol. 4, (2005).

[8] Alex Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385, Springer, 2012.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', *CoRR*, **abs/1512.03385**, (2015).

[10] Zhanfeng Jia, Chao Chen, B. Coifman, and P. Varaiya, 'The pems algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors', in *Proceedings of IEEE Intelligent Transportation Systems.*, pp. 536–541, (2001).

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, 'Imagenet classification with deep convolutional neural networks', in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS, pp. 1097–1105, (2012).

[12] Hoang Thanh Lam, Ernesto Diaz-Aviles, Alessandra Pascale, Yiannis Gkoufas, and Bei Chen, '(blue) taxi destination and trip time prediction from partial trajectories', in *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge*, pp. 63–74, (2015).

[13] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu, 'Multi-task representation learning for travel time estimation', in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pp. 1695–1704. ACM, (2018).

[14] Zhongjian Lv, Jiajie Xu, Kai Zheng, Hongzhi Yin, Pengpeng Zhao, and Xiaofang Zhou, 'Lc-rnn: A deep learning model for traffic speed prediction', in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3470–3476, (7 2018).

[15] Xiaolei Ma, Zhuang Dai, Zhengbing He, and Yunpeng Wang, 'Learning traffic as images: A deep convolution neural network for large-scale transportation network speed prediction', *CoRR*, **abs/1701.04245**, (2017).

[16] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang, 'Long short-term memory neural network for traffic speed prediction using remote microwave sensor data', *Transportation Research Part C: Emerging Technologies*, **54**, 187 – 197, (2015).

[17] B. Pan, U. Demiryurek, and C. Shahabi, 'Utilizing real-world transportation data for accurate traffic prediction', in *IEEE 12th International Conference on Data Mining*, pp. 595–604, (2012).

[18] J. Rice and E. van Zwet, 'A simple and effective method for predicting travel times on freeways', *IEEE Transactions on Intelligent Transportation Systems*, **5**(3), 200–207, (Sept 2004).

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, 'Sequence to sequence learning with neural networks', in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS, pp. 3104–3112, (2014).

[20] Hongjian Wang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li, 'A simple baseline for travel time estimation using large-scale trip data', in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 61:1–61:4. ACM, (2016).

[21] Zheng Wang, Kun Fu, and Jieping Ye, 'Learning to estimate the travel time', in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pp. 858–866. ACM, (2018).

[22] Chun-Hsin Wu, Jan-Ming Ho, and Lee D.T., 'Travel-time prediction with support vector regression', *IEEE Transactions on Intelligent Transportation Systems*, **5**(4), 276–281, (2004).

[23] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee, 'Travel-time prediction with support vector regression', *IEEE Transactions on Intelligent Transportation Systems*, **5**(4), 276–281, (Dec 2004).

[24] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang, 'Modeling trajectories with recurrent neural networks', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3083–3090, (2017).

[25] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang, 'Modeling trajectories with recurrent neural networks', in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 3083–3090, (2017).

[26] Tao Xu, Xiang Li, and Christophe Claramunt, 'Trip-oriented travel time prediction (tottp) with historical vehicle trajectories', *Frontiers of Earth Science*, **12**(2), 253–263, (Jun 2018).

[27] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, 'Spatial-aware hierarchical collaborative deep learning for poi recommendation', *IEEE Transactions on Knowledge and Data Engineering*, **29**(11), 2537–2551, (2017).

[28] Hanyuan Zhang, Hao Wu, Weiwei Sun, and Baihua Zheng, 'Deeptravel: a neural network based travel time estimation model with auxiliary supervision', in *IJCAI*, (2018).

[29] Junbo Zhang, Yu Zheng, and Dekang Qi, 'Deep spatio-temporal residual networks for citywide crowd flows prediction', in *AAAI*, (2017).

[30] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi, 'Dnn-based prediction model for spatio-temporal data', in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS, pp. 92:1–92:4. ACM, (2016).

[31] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang, 'Urban computing: Concepts, methodologies, and applications', *ACM Trans. Intell. Syst. Technol.*, **5**(3), 38:1–38:55, (2014).