

Dynamic Model for Traffic Flow Prediction Using Updated Deep Residual Network

Zeren Tan^a and Ruimin Li^b

^aDepartment of Civil Engineering, Tsinghua University, Beijing 100084, China

^bInstitute of Transportation Engineering, Tsinghua University, Beijing 100084, China*

May 2018

Abstract

Real-time traffic flow prediction can provide travelers with reliable traffic information to save people's time, and also assist traffic operation agencies in managing the traffic system effectively. In this study, a dynamic prediction model was established by using an improved deep residual network (DRN). On the basis of the traditional DRN model, we first integrated the input and output of the i^{th} layer to the input of the $(i + 1)^{th}$ layer and proved that each layer fit a simple function thereby reducing the error rate as the model was trained rapidly. Then, we used the concept of on-line learning on our model to update the pretrained model during the prediction process and rebuild our model with ongoing time series. The influence of the number of layers in our network was investigated by conducting experiments. Results verified that performance was not necessarily improved when the network increased in depth. We also evaluated our network on both short and long time intervals. Result showed that our dynamic model demonstrated better accuracy than other state-of-the-art models when applied to traffic volume prediction in an expressway in Beijing. In addition, our dynamic model can perform better in practical applications.

Keywords: Traffic flow prediction, Deep learning, Deep residual network, Dynamic model

1 Introduction

Precise, rapid and timely traffic flow prediction is a major task of intelligent transportation systems (ITSs) (Xingyuan Dai, 2017). Making decisions based on real-time and predicted traffic flow is important to travelers, companies, and governments (Chen and Englund, 2016). Office workers need this type of information to schedule trips to their workplace. Traffic operation agencies can use the predicted traffic flow information to plan vehicle lanes (Yuan et al., 2011; Wu et al., 2018). However, despite

the continuous research and development worldwide, accurate and short-term traffic flow prediction has remained a challenge to researchers for decades because of its stochastic and nonlinear characteristics. At the beginning of traffic flow prediction research, experts have mainly used linear methods, such as autoregressive integrated moving average (ARIMA) (Ahmed and Cook, 1979). These linear models, such as multivariable linear regression (MVLRL) (L. Li and Zhang, 2015), have been improved continuously and applied in traffic flow prediction, due to their simplicity and convenience. Despite the progress in linear models, prediction accuracy for nonlinear traffic flow still needs improvement. Therefore, finding powerful methods to predict accurate traffic flow is important for researchers.

From approximately 40 years ago, machine learning algorithms have demonstrated good performance in many tasks such as prediction and classification. Thus, researchers have begun to use machine learning algorithms such as support vector regression (SVR) (X. Jin and Yao, 2007) and k-nearest neighbor (k-NN) (Davis and Nihan, 1991), in traffic flow prediction. In spite of the increase in performance, these approaches fail to consider all characteristics of the traffic flow and obtain unsatisfactory performance due to the complexity of traffic flow. Thus, traffic flow prediction has attracted considerable attention from researchers from civil engineering, automation, industrial engineering, and computer science.

In recent years, deep learning methods, which are a subset of machine learning methods, have shown remarkable performance in many computer science-related tasks, such as image classification, machine translation and speech recognition. An increasing number of researchers working in other fields, including civil engineering, have been inclined to use deep learning methods in structural damage detection (Lin et al., 2017; Cha et al., 2017a,b; Maeda et al., 2018; Gao and Mosalam, 2018), concrete defect detection (Li et al., 2018), pavement crack detection (Zhang et al., 2017), transportation network reliability (Nabian and Meidani, 2018), traffic speed estimation (Liu et al., 2018), and real-time transportation management (Hashemi and Abdelghany, 2018). With the recent development in deep learning, traffic flow prediction has become an important research direction in the application of deep learning approaches, such as Recurrent Neural Network (RNN) (Kyunghyun Cho and Bengio, 2014), Long Short-Term Memory network (LSTM) (Tian and Pan, 2015), Gated Recurrent

*Email: lrmin@tsinghua.edu.cn

Unit (GRU) (Fu et al., 2016), Stacked Auto-Encoders (SAEs) (Y. Lv and Wang, 2014), Deep Belief Network (DBN) (Huang et al., 2014), and some other models (Vlahogianni et al., 2008; Stathopoulos et al., 2008; Ghosh et al., 2010; Tan and Li, 2018; Song et al., 2018). These models generally have complex network architectures that can capture the nonlinear nature of traffic flow. Hence, previous studies showed that these methods perform better in forecasting traffic flow than the traditional models in their experiments (Fu et al., 2016; Tian and Pan, 2015; Y. Lv and Wang, 2014; Huang et al., 2014; Kyunghyun Cho and Bengio, 2014).

RNN (Kyunghyun Cho and Bengio, 2014), LSTM (Tian and Pan, 2015) and GRU (Fu et al., 2016) are commonly used deep learning models. However, a considerable amount of time is required in training these models due to their complicated architectures. Experts are often required to stack less layers and set training epochs to a small number, thereby reducing accuracy to a certain extent. Researchers can also use other techniques such as dropout to reduce the size of their training set, which will lead to a lower accuracy when the data set is not large enough. Therefore, using these models in practical applications is difficult. Previous studies have been based on static models, that is, models are not updated after training. However, realistic traffic flow condition is continuously updated and includes significant uncertainty. In addition, traffic flow data are periodic, with 1 day and 1 week as its periods (Tan et al., 2013, 2016; Wu et al., 2017). Furthermore, traffic flow is uncertain and changes all the time (Chen et al., 2018). Therefore, we need to build a prediction mechanism for real-time updates and adapt to the continuously changing traffic flow status.

In this study, we improved the traditional deep residual network (DRN) (He et al., 2016) and proposed a Dynamic Improved Deep Residual Network (DIDRN) based prediction framework that will continuously update its training set when real data are available, and perform well in practical applications. The results showed that our model demonstrated robust performance in multistep prediction with a short amount of training time.

The rest of this paper is organized as follows. Section 2 reviews the existing literature on short-term traffic flow prediction. Section 3.1 demonstrates the basic concepts of DRN. Section 3.2 and Section 3.3 explain our process of further updating DRN and designing the dynamic model, respectively. Section 4 presents experiments, compares our model's performance with commonly used models and analyzes the performance of our model in detail. Finally, section 5 draw the conclusions and discusses future studies. Appendix lists all the algorithms used in this research.

2 Literature Review

Traffic flow prediction is a main task of ITSs. Research on traffic flow prediction has progressed considerably over the years. New models are constantly being proposed as the performance of these models also increases at the same time.

Traffic flow prediction has time-series (Cheng et al., 2012), regression (Dunne and Ghosh, 2011) or clustering problems (Xia et al., 2012). As methods improve, classical statistical inference methods have been replaced by alternative models such as Bayesian inference, multivariate time series models and neural

networks (Vlahogianni et al., 2004).

The selection of traffic flow prediction model is an issue in traffic flow prediction (Abdulhai et al., 1999). Existing traffic flow prediction methods can be generally divided into the two main categories: of parametric methods non-parametric methods.

Parametric methods such as ARIMA (Ahmed and Cook, 1979), Kalman-filtering (Chen and Grant-Muller, 2001; Chien and Kuchipudi, 2003; Chien et al., 2003; Wang et al., 2006; Van Lint, 2008; Jin et al., 2013; Guo et al., 2014) and MVLR (L. Li and Zhang, 2015) are often used by researchers. These approaches require predetermined model architecture and that the parameters of the model are calculated by using empirical data (Xingyuan Dai, 2017). Improved models based on ARIMA such as SARIMA were proposed in the 21st century (Williams, 2001; Smith et al., 2002; Billy M. Williams and Lester A. Hoel, 2003; Williams and Hoel, 2003; Chandra and Al-Deek, 2009). These models have simple and straightforward architectures but require a considerable amount of data and that the traffic condition is in a stationary process. Hence, these approaches are unsuitable when sufficient data are not available and their performance is unsatisfactory when the detected traffic flow of a certain location is not stable enough. In addition, parametric models require prior knowledge on finite sets of parameters. Thus, setting these parameters is a challenge. Moreover, traffic flow data are random and uncertain. Therefore, those parametric approaches cannot perform well in the majority of these cases. They are also limited both theoretically and practically.

The structure and parameters of nonparametric models need not to be fixed in advance (Ma et al., 2015). At the beginning of the 21st century, researchers have focused on non-parametric approaches such as k-NN (Davis and Nihan, 1991), Bayesian Network approach (Sun et al., 2006a; Castillo et al., 2008; Sun et al., 2006b), SVR (X. Jin and Yao, 2007; Wu et al., 2004; Asif et al., 2014), random forests regression (RF) (Leshem and Ritov, 2007), gradient boosting regression (Friedman, 2001). These machine learning approaches also require a large amount of data and may not be the optimal choice when sufficient data are not available (Davis and Nihan, 1991; Sun et al., 2006b). As Sun et al. (2006b) mentioned in their paper, their Bayesian network model can work well when data are incomplete but it is not the best choice to use their model when some abnormal scenarios occur. Zheng et al. (2006) introduced a Bayesian combination method (BCM) (Petridis et al., 2001) for traffic flow prediction. However, as Wang et al. (2014) pointed out, BCM does not consider the relevance between historic traffic flows and current traffic flows. Therefore, this model is not recommended for traffic flow prediction because traffic flow is a type of sequential data thereby indicating that current data are related to historical data. However, researchers from many fields, including transportation, have used Artificial Neural Networks due to their excellent performance in capturing nonlinearities and its more flexible structure, generalization ability, learning ability and adaptability (Li and Lu, 2009; Li and Rose, 2011; Karlaftis and Vlahogianni, 2011; Ma et al., 2015).

Applications of neural networks in traffic flow prediction have been extended largely from simple multilayer perceptrons (Clark et al., 1993), to advanced structures, such as time-delayed recurrent neural networks (Abdulhai et al., 1999).

With the rapid development of deep learning, many deep

learning models have been proposed by computer scientists. This type of model can represent the complex features of traffic flow without prior hypothesis (Lv et al., 2015; Chen et al., 2018). Developing a neural network model by adding layers and feeding your data into the network are required. Prior knowledge is unnecessary in depth when training and predicting are conducted. However, theoretical evidence shows that a single-layer neural network can represent any Borel measurable function (Hornik et al., 1989). Many powerful deep learning models, such as SAE (Y. Lv and Wang, 2014), DBN (Huang et al., 2014), RNN (Kyunghyun Cho and Bengio, 2014), LSTM (Tian and Pan, 2015; Ma et al., 2015), GRU (Fu et al., 2016; Zhang and Kabuka, 2018) and Fuzzy Deep Convolution Neural Network (FDCN) (Chen et al., 2018), were introduced in traffic flow prediction and showed superior performance compared with traditional models. Y. Lv and Wang (2014) proposed a greedy layer-wise unsupervised learning algorithm to train the model thereby reducing the training time significantly without accuracy loss. Tian and Pan (2015) proposed a model called LSTM RNN for short-term traffic flow prediction and their results achieved higher accuracy than classic models such as SVM. Fu et al. (2016) proposed GRU NN for traffic flow prediction and their results showed that GRU and LSTM outperformed ARIMA. Chen et al. (2018) developed a fuzzy deep learning model called FDCN. Apart from demonstrating powerful representing capability, their results also showed stability in terms of root mean square error (RMSE) in performing multistep forecasting.

Sequential data are correlated with data in the context. However, common neural networks cannot capture this correlation among data. In the 1990s, RNN (Hochreiter, 1991) was developed to solve this problem. Although RNN can capture correlation among data, this correlation is soon forgotten, that is, this neural network does not have long-term memory (Zhao et al., 2017; Ma et al., 2015). Hence, if data are strongly correlated and historical data have a remarkable influence on consequent data even after a long time, choosing RNN as the training model is not recommended. In 1997, long short-term memory (LSTM) networks (Sepp Hochreiter, 1997) were proposed and set accuracy records in various application domains. LSTM can remember correlation for a long time and thus demonstrates outstanding performance in sequential modeling. However, both RNN and LSTM are difficult to train because the architecture of these models is too complex (Lei and Zhang, 2017). Researchers often take hours or even days to train these models so training model is time-consuming especially when the dataset is very large. Despite their good performance, using them in practical scenarios is sometimes impossible. To reduce mathematical operations and total running time, gated recurrent unit (GRU) was developed by Fu et al. (2016). Unfortunately, in practice, although GRU can reduce training time, the cost is still high. In addition, Klaus Greff (2017) successfully compared variants of LSTM by conducting a large-scale study on many different datasets and found that these LSTM based methods are all the same and none of its modifications improved performance significantly.

Apart from RNN and LSTM, researchers have also used some other deep learning models. Gang Song (2017) proposed a novel double deep ELMS (Extreme Learning Machines) ensemble system (DD-ELMs-ES) which used three deep ELM models as basic models and their model demonstrated better generalization

performance than some state-of-the-art algorithms. However, their model only focused on one-step forecasting. Xingyuan Dai (2017) implemented DeepTrend which decomposed original traffic flow data into trend and residual components. They demonstrated that DeepTrend could noticeably improve prediction performance and outperform many traditional models as well as LSTM. Xiaochuan Sun (2017) proposed a deep belief echo-state network (DBEN) to address the problem of slow convergence and local optimum in time series prediction. Their experimental results demonstrated that DBEN has good performance in learning speed and short-term memory capacity. However, a large number of parameters exist in their model, thereby indicating difficulty in parameter tuning.

He et al. (2016) proposed deep residual network (DRN) in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Their model substantially outperformed other models and won the first prize in this competition. Typically, training difficulty increases as the number of layers also increases in network, and the problem of gradient vanishing will arise in the mean time. Their model performed training easily and solved this problem effectively even when the deep neural network is several times deeper than other networks. For a normal deep neural network, the maximum depth that people have been less than 100. However, for DRN, a network with several hundred layers can be easily created and training can be conducted in a short time. Thus, we were motivated to apply DRN concepts to traffic flow prediction. Accordingly, we improved the architecture of DRN and its effectiveness. Section 4 illustrates our results.

The main contributions of this study are as follows:

- We apply and improve deep residual network in traffic flow prediction.
- We consider practical applications and propose a dynamic model called DIDRN.
- Results show that our model is more powerful than other commonly used models.

3 Model Development

In this section, we first introduced the concept of a normal neural network works. We then explained the structure of DRN and quoted a hypothesis proposed in (He et al., 2016). In the final subsection, the structure of DIDRN was proposed.

3.1 Original DRN

When the popularity of deep learning began to increase, experts simply stacked layers and assumed that deeper is better. The network was given an input x and the intermediate layers fit a map H to obtain the final output $H(x)$. As they believe deeper networks could have better performances, they stack more layers to fit the desired map H and expect to obtain a higher accuracy. However, when the depth of the network increased, both training and test accuracy get saturated, that is, degradation occurred (He et al., 2016). Figure 1 shows the architecture of a simple stacked neural network.

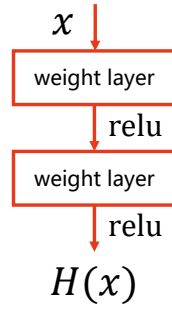


Figure 1: Architecture of common neural network

DRN was proposed to solve degradation problem. Instead of expecting that each stacked layer will fit into a desired mapping directly. These layers fit the function via residual mapping.

Hence, assuming that the network is expected to fit $H(x)$, He et al. (2016) let the stacked layer fit another mapping $F(x) = H(x) - x$. Then the desired mapping $H(x)$ could be replaced by $F(x) + x$.

They hypothesized that fitting the residual mapping $F(x)$ was easier than the original mapping $H(x)$. If the optimal mapping is identity, then it would be easy to make the residual be zero. Figure 2 shows the idea stated above.

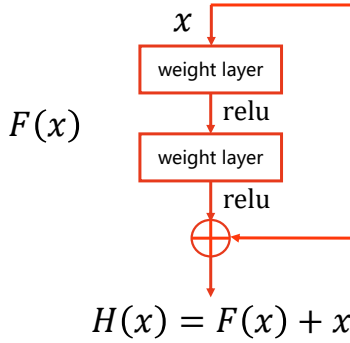


Figure 2: Architecture of residual neural network

This simple change made a remarkable difference as training the network becomes easy. Deep residual network has helped Kaiming et.al. win first prize in the ILSVRC classification competition and many other competitions. Prior to DRN, the award-winning network of ILSVRC had a maximum number of 32 layers. However, when DRN participated in this competition, it had 152 layers with a significant increase in accuracy.

3.2 Updated DRN

In the training process of neural networks, the estimated function is derived from the objective function to achieve computation ac-

curacy. Therefore, the accuracy will decrease due to information loss. To preserve more information when information is transferred among layers and thus obtain high accuracy, we further improved the architecture of DRN and used it in traffic flow prediction. Figure 3 shows the architecture of our improved network.

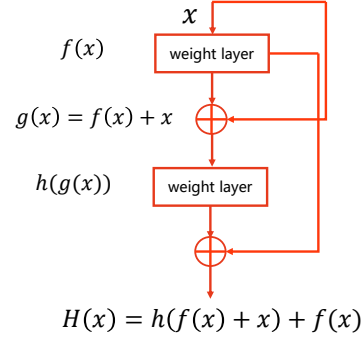


Figure 3: Our improved residual network

The training process of our improved network is as follows:

- Denote the input as x , and the first layer fits the function $f(x)$.
- Add these two outputs together to obtain $g(x) = f(x) + x$.
- The second layer considers $g(x)$ as the input and maps it into $h(g(x))$.
- Similarly, add the output of the first layer $f(x)$ and the output of the second layer $h(g(x))$ to obtain $H(x) = h(f(x) + x) + f(x)$.

From these steps, the task of fitting a complex function $H(x)$ was divided into fitting two simpler mappings f and h , respectively. According to the hypothesis of He et al. (2016), the layers will work more efficiently because fitting these two simple mappings is easy, thereby reducing the error rate in fitting these two mappings and subsequently reducing the overall error rate.

3.3 Dynamic Model

In practice, a traffic operation center would not use this model in the exact place where our training data come from, that is, our model needs to be transferred easily among different places. In addition, traffic flow on a specific road could change over time. Hence, the traffic flow after 1 month or 1 year may have significant difference with the existing situation, especially in rapidly developing cities. In the DRN model, weights and parameters were obtained by using the previous training data. These weights and parameters could fit the previous data well but it may not fit the new data well, that is the previous model may not perform well by using the new data. Therefore, simply using the existing model with only one training process to predict traffic flow without any improvement was unreasonable. The old model could not be expected to perform well when the traffic flow changed

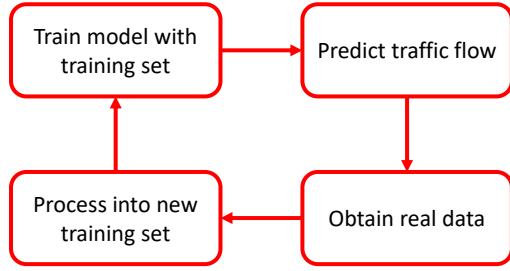


Figure 4: Dynamic Model

significantly. Hence, traffic flow prediction models should be updated constantly. On the basis of these ideas, we designed a dynamic model based on the improved DRN.

We used the basic idea of incremental learning to update the pretrained model and implement the dynamic model.

The complete process is as follows:

- Step 1 We use the collected data to pretrain the basic mode 1.
- Step 2 We use our model to predict traffic flow.
- Step 3 After some time, the basic model complete the predict ion steps, and we obtain some new real data.
- Step 4 We combine these new data with the existing training set into a new training set and use this new training set to retrain ou r model.
- Step 5 Repeat Step 2-Step 4.

Figure 4 shows this process.

With these steps, the model will be continuously updated by absorbing new data. Therefore, the proposed dynamic model satisfied the practical conditions better and obtained more accurate predictions than the existing model.

3.4 Model Implementation

In this section, we will present the entire process of implementing the model in detail.

The process is divided into the following steps:

- Step 1 Denote the raw data as R , which is a column vector of R^n .

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix}$$

Assume that trend characteristics exist in traffic flow data. Thus, we subtract the adjacent data to obtain the difference vector $D = [D_1, D_2, \dots, D_{n-1}]$, where

$$D_i = R_{i+1} - R_i \quad i = 1, 2, \dots, n-1$$

- Step 2 However, we cannot directly apply supervised learning to D . Hence, we further process our data into supervised data.

First, select the time-step, which represents the number of previous data points that were used to predict the next data point. After parameter tuning, we choose 1 as our time-step, that is, time-step = 1.

Then we denote X as the feature and Y as the data to be predicted. X and Y are defined as follows.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

where

$$X_1 = 0 \quad X_i = D_{i-1}, i = 2, 3, \dots, n-1$$

$$Y_i = D_i, i = 1, 2, \dots, n-1$$

Then we obtain the entire data set as follows:

$$\begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ \vdots & \vdots \\ X_{n-1} & Y_{n-1} \end{bmatrix}$$

- Step 3 Normalization, which is a simple approach often used in developing deep learning models and performing experiments, is conducted to accelerate training and prediction. Elements in D can be positive or negative. Thus, X and Y are scaled into $(-1, 1)$. We define the following obtained vectors as $scaled_X$ and $scaled_Y$.

$$scaled_X = \begin{bmatrix} scaled_X_1 \\ scaled_X_2 \\ \vdots \\ scaled_X_{n-1} \end{bmatrix} \quad scaled_Y = \begin{bmatrix} scaled_Y_1 \\ scaled_Y_2 \\ \vdots \\ scaled_Y_{n-1} \end{bmatrix}$$

where

$$scaled_X_i = 2 \frac{X_i - X_{min}}{X_{max} - X_{min}} - 1$$

$$scaled_Y_i = 2 \frac{Y_i - Y_{min}}{Y_{max} - Y_{min}} - 1$$

and

$$scaled_X_i \in (-1, 1), i = 1, 2, \dots, n-1$$

$$scaled_Y_i \in (-1, 1), i = 1, 2, \dots, n-1$$

Then the data set is transformed as:

$$\begin{bmatrix} scaled_X_1 & scaled_Y_1 \\ scaled_X_2 & scaled_Y_2 \\ \vdots & \vdots \\ scaled_X_{n-1} & scaled_Y_{n-1} \end{bmatrix}$$

- Step 4 During prediction, we invert Step 1 and Step 3 to obtain the final prediction data.

All algorithms (Algorithm 2 ~ Algorithm 5) used to implement the entire process are presented in Appendix A

4 Experiments and Results

4.1 Data Source

Our data were obtained from the traffic detectors located in the ring roads of Beijing, China. Basic traffic flow data are collected via 14 microwave detectors, which were numbered as 2010, 2011, 2013, 2023, 2030, 2033, 2052, 3034, 3035, 4004, 4005, 4050, 4051, and 5062. The time interval in these traffic flow data was 10 minutes. The time span is 61 days from June 1st, 2013 to July 31st, 2013. Therefore, there are 8784 data points for each detector. For data from each detector, we use the first 7200 pieces of the entire data set as training data and the rest as test data. The detected traffic flow is loaded as input and finally we get the predicted traffic flow at specified time points as output. The results verify the feasibility and effectiveness of our proposed model.

4.2 Performance Indexes

Rooted Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE) are chosen as performance indexes to evaluate the performance of these four models.

RMSE, MAPE and MAE were defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$
$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|\hat{y}_i - y_i|}{y_i} \times 100\%$$
$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

where y_i and \hat{y}_i refer to the real and forecasted values.

We use these three performance indexes to evaluate the following aspects of the selected models: 1) absolute error and 2) relative error. RMSE and MAE represented the deviation from the predicted values and the detected values. They also increase with the increase in the range of data. MAPE measured the deviation from the predicted values and the ground truth values. This index determined percentage error, which was not related to the range of our data.

4.3 Performance Analysis

To analyze the efficiency of our model, the following experiments have been carried out.

- The performance of DIDRN in terms of the different number of layers was inspected to determine the changes in prediction accuracy when the number of layers changed as shown in Section 4.3.1.
- The performance of different models in short-term traffic flow prediction was explored as shown in Section 4.3.2.
- The performance of DIDRN under different prediction time intervals was investigated, as shown in Section 4.3.3. Concretely, we increase the time interval gradually from 10 minutes to 7 days to evaluate the stability of our model, analyze

its performance and determine the optimal model in predicting traffic flow.

The experiments were performed on a laptop with @2.60 GHz processor, 8.0 GB RAM and NVIDIA GeForce GTX 960M. The models were coded via Python 3.6 with Keras and TensorFlow frameworks and compiled by using the Anaconda Jupyter Notebook.

4.3.1 Impact of the Number of Layers

First, the input dimension should be determined, that is the number of data points we input to the models is considered one input point. For example, if the input dimension was chosen as 5, then the models used 5 data points to predict the traffic flow of next time point. After parameter tuning, the input dimension was chosen as 1.

As previously mentioned, when the depth of the network increases, the accuracy rate will first increase and then decrease. To verify this, experiments were performed on DIDRN with different numbers of layers. In these experiments, the chosen time interval was 10 minutes. Figure 5 shows that a shallow network obtained relatively good performance, but a deep network increases the error rate. When the network had over 40 layers, the error rate increased significantly. We selected small numbers and found that the error rate was the lowest among all experimented numbers when the number was equal to 16. Although a similar error rate was still obtained when the number was 36, the training time became longer as the network became deeper. When the number of layer exceeded 40, the error rate exhibited a clear trend of increasing and that was significantly higher than that when the network was shallow. Hence, we choose less points in our experiments and the result thoroughly verify the reported problem in (Srivastava et al., 2015; He and Sun, 2015; He et al., 2016).

4.3.2 Short-term prediction

To evaluate the efficiency of our proposed model, we performed several experiments on our proposed model and other similar models, including the one layer LSTM, deep LSTM, and DRN. From the result in Section 4.3.1, except for those of the one layer LSTM, the three other models have 16 layers. To change the input dimension into the output dimension, we added some layers to both DRN and DIDRN but maintained both should still have 16 layers in total.

We first used these models to predict the short-term traffic flow. Concretely, the interval of time points was 10 minutes. That is, we used the previous traffic flow to predict the traffic flow 10 minutes later.

Table 1 and Figure 6 show the performance.

Table 1 showed that DIDRN obtained performance than the other models. One layer LSTM obtained the worst performance among the models because of its shallow depth. Deep LSTM achieve satisfactory performance. However, on our machine, it actually takes nearly 5 minutes to process data from one detector and train it before it converges. DRN, which was significantly easier to train, obtained a similar but lower error rate than deep LSTM. DRN only takes us about 50 seconds to train. By contrast, although DIDRN was derived from DRN, it achieved a lower error rate than DRN, thereby indicating that our model was efficient

Table 1: Performance comparison of different models

Model		One layer LSTM	Deep LSTM	DRN	DIDRN
2010	RMSE	101.07	75.65	75.64	75.04
	MAPE	11.58%	7.05%	7.04%	6.43%
	MAE	84.60	59.13	59.12	56.43
2011	RMSE	123.79	104.13	104.13	102.89
	MAPE	11.15%	7.90%	7.90%	7.57%
	MAE	98.61	77.70	77.71	76.33
2013	RMSE	122.05	96.47	96.37	94.88
	MAPE	15.56%	9.46%	9.43%	8.88%
	MAE	98.82	71.23	71.13	69.61
2023	RMSE	99.99	78.28	78.02	74.26
	MAPE	11.19%	7.02%	6.97%	6.96%
	MAE	81.86	58.90	58.60	55.72
2030	RMSE	104.95	76.00	75.96	67.13
	MAPE	14.63%	7.43%	7.42%	7.38%
	MAE	86.94	57.34	57.29	49.66
2033	RMSE	79.84	59.75	59.75	58.40
	MAPE	12.63%	6.89%	6.89%	6.80%
	MAE	66.70	45.11	45.11	44.11
2052	RMSE	109.56	85.99	85.99	83.28
	MAPE	11.63%	7.46%	7.46%	7.20%
	MAE	88.31	65.52	65.52	62.77
3034	RMSE	99.08	74.32	74.04	72.27
	MAPE	10.45%	6.97%	6.89%	6.89%
	MAE	83.05	57.07	56.99	53.67
3035	RMSE	98.09	68.36	68.36	64.74
	MAPE	15.15%	7.64%	7.64%	7.59%
	MAE	82.82	51.15	51.15	46.68
4004	RMSE	94.20	61.89	61.89	57.30
	MAPE	16.44%	8.21%	8.21%	6.80%
	MAE	80.25	47.01	47.01	42.39
4005	RMSE	106.9	87.65	87.65	86.02
	MAPE	10.02%	6.34%	6.34%	6.27%
	MAE	87.13	65.63	65.63	64.35
4050	RMSE	93.53	57.85	57.85	56.48
	MAPE	37.96%	19.36%	19.36%	18.58%
	MAE	80.57	37.89	37.89	36.11
4051	RMSE	112.35	98.13	98.13	98.60
	MAPE	12.04%	8.66%	8.66%	8.45%
	MAE	83.94	65.56	65.56	65.20
5062	RMSE	90.63	60.06	60.06	56.18
	MAPE	17.89%	8.23%	8.23%	8.06%
	MAE	77.86	45.87	45.87	42.24

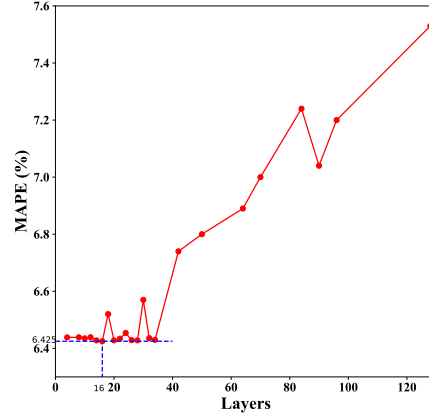


Figure 5: Impact of number of layers

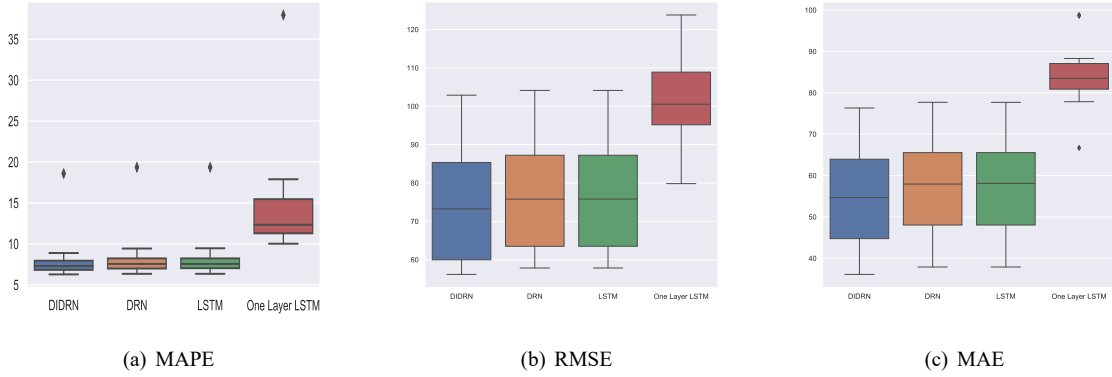


Figure 6: Forecast Error of Different Models

enough and training was easy. We could complete the training process within 100 seconds which was slightly longer than that of DRN but was significantly shorter than that of deep LSTM. DRN and Deep LSTM obtained similar performance results because both are comparably accurate when the network is not deep enough (He et al., 2016). This result was similar to those demonstrated in their study. When the network became very deep, their variation would be revealed. But generally, DRN needs less time to be trained.

Table 1 presents that the model accuracy of dataset 4050 is significantly lower than those of other datasets. We plot the data points of dataset 2010 and dataset 4050 as shown in Fig 4.3.2. The shadow represented the range of traffic flow at each time point and the solid line denoted the mean traffic flow value of that time point. The figure showed that the more discrete data in dataset 4050 was not as stable as those of other dataset. The evaluated models generally obtained poor performance in this type of circumstance.

Figure 7 shows the comparison of predicted data and measured data in DIDRN and dataset 2010, respectively.

4.3.3 Further Analysis on Performance of DIDRN

The experiments were conducted under the following motivations:

- (1) Our model showed good performance when the chosen time interval was 10 minutes. However, a model commonly obtains poor performance when the time interval is enlarged. We were motivated to perform an in-depth investigation on the impact of time interval on model performance.
- (2) Table 1 shows that when time interval is 10 minutes, the MAPE is satisfactory. However, we aimed to determine experimentally in this section if our model can obtain even better performance by using another time interval.
- (3) The practical scenario was far more complicated than the theoretical one. A problem occurred when our model obtained the best performance theoretically at the time interval of 10 minutes. However, in practice, the detector could only provide us the data with a time interval larger than 10 minutes. The third motivation in this section led to the following questions: how did our model perform by using that data? Can



Figure 7: Measured data and predicted data

we use the data obtained from 1 day or 1 week ago if these are the only data available? What is the accuracy of our model under these conditions.

In the following discussion, our analysis was conducted mainly based on MAPE because of its simplicity and intuitiveness. The data of detector 2010 were chosen to demonstrate the results by using random selection. However, data from either detectors would show similar results.

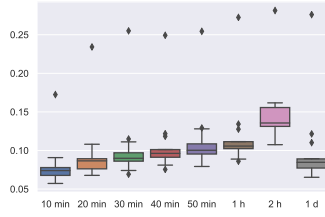
To evaluate the stability of our proposed model, we changed the time step from 10 minutes to 1 hour with a time interval of 10 minutes, 2 hours, and 24 hours. Moreover, DIDRN was used to conduct the experiments.

Table 2 shows that accuracy decrease monotonously when the time interval increases from 10 minutes to 1 hour and 2 hours, the accuracy decreases monotonously. This result is congruent with our intuition. When the time interval became larger, we needed more information to predict the traffic flow and the traffic flow of the previous time point is less related to the traffic flow

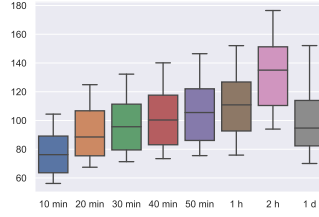
of the predicted time point. Therefore, accuracy was reduced. However, when we use the traffic flow of the previous day to predict that of the next day, accuracy significantly increased to 10.88% in terms of MAPE. On one hand, traffic flow data are stable with a periodic nature (Tan et al., 2013, 2016; Wu et al., 2017). Thus, the traffic flow of the same time in the previous day was relatively similar to that of the predicted day, especially when two consecutive workdays were considered. On the other hand, traffic flows after 1 or 2 hours could be significantly different with those of the current time point. For example, the traffic flow at 7:00 a.m. was significantly different compared with that at the peak hours of 8:00 a.m. and 9:00 a.m. People go to work during peak hours, thereby leading to a sudden increase in traffic flow. As the only input was data from 7:00 a.m. when the model attempted to predict the traffic flow at 8:00 a.m. or 9:00 a.m., the model did not have enough information to predict this sudden increase. Hence, the predicted values obtained a larger deviation from the detected values.

Table 2: Performance of DIDRN on Different Time Intervals

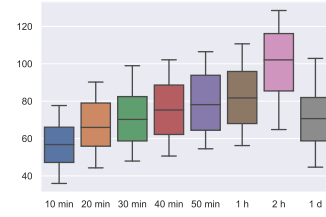
Time Interval (minutes)	RMSE	MAPE	MAE
10	75.04	6.43%	56.43
20	119.39	11.58%	91.77
30	128.11	11.35%	93.01
40	198.34	19.61%	153.16
50	197.69	17.26%	140.90
60	202.87	18.45%	141.34
120	342.96	32.67%	233.05
1440	135.24	10.88%	94.92



(a) MAPE



(b) RMSE



(c) MAE

Table 3 lists the MAPE values of different detectors in terms of predicting 1 day in advance. Hence, we used the data of the previous days to predict the traffic flow of the next day. The results revealed that the model obtained a stable performance from data of different detectors. The model achieved poor performance from data of detector 4050 because the corresponding data of this detector were not fluctuant. That is, the traffic flow obtained by the detector in this location was unstable over time. Our model could not be expected to perform well when the data were not good and stable enough. This observation was supported by the results shown in Table 1.

To further analyze the impact of time interval, we gradually increased the time interval from 1 to 7 days.

Table 4 shows the results. When the time interval ranged from 2 to 6 days, the error rate ranged from 13.96% to 10.45%. Because the predicted traffic flow had reduced relation with the input traffic flow when the time interval increased. The model had less information to update its weights and perform well in both training set and test set.

A large increase occurred between 6 and 7 days. Owing to symmetry, the error rate of 6 days should be similar to that of 1 day, which was also consistent with our results. As traffic flow data are a type of periodic data and the period was exactly 7 days, it is not surprising that the flow 1 week later is similar to the current flow. Owing to this property, our model obtained better performance and higher accuracy than the others.

We increased the time interval from 10 minutes to 2 hours and compared the performance with the presented results. In this study, the kind of time interval, which the performance is better when time interval is less than it and the performance is worse when time interval is greater than it, is referred as boundary time (BT). Figure 8 shows the results of time intervals at less than 2 hours, 1 day and 7 days. This figure illustrated that MAPE in-

creased when the time interval increased. The best performance was obtained by our model when time interval is equal to 10 minutes. The performance of 1 day was better than that of 30 minutes but was worse than 20 minutes, whereas the performance of 1 week was better than that of 20 minutes but was worse than that of 10 minutes. This experiment was also conducted on data from the other detectors. Table 5 lists the boundary time for all detectors.

On the basis of these results, the following conclusions could be drawn:

1. If data of the previous time point were available, then using those data to predict traffic flow of next time point was the best option. That is, the time interval should be set to 10 minutes.
2. If data of the previous week were available or the task was to conduct long-term prediction, then using these data was better.
3. When the time interval was greater than 30 minutes, the performance was generally better than 1 day or 1 week time intervals. In practical applications, BT could be obtained by using historical data, which could be used as a reference for choosing the time interval in predicting the traffic flow.

5 Conclusion and Future work

In this study, we demonstrated the basic concepts of deep residual network. The results showed that DRN was much simpler to train and obtained better performance than other approaches. We then explain the motivations and improvements of DRN in our research. We showed the architecture of our network and how

Table 3: One day forward prediction accuracy of different detectors

Detector	2010	2011	2013	2023	2030	2033	2052
MAPE	10.88%	12.64%	15.01%	10.66%	10.74%	9.43%	12.31%
Detector	3034	3035	4004	4005	4050	4051	5062
MAPE	10.71%	10.50%	11.35%	9.83%	52.40%	33.60%	12.27%

Table 4: Performance of DIDRN for weekday and weekend

Days	RMSE	MAPE	MAE
1	135.24	10.88%	94.92
2	158.27	13.96%	113.70
3	154.55	13.81%	109.36
4	145.62	13.22%	104.54
5	142.73	10.85%	99.14
6	143.36	10.45%	98.12
7	100.56	7.28%	69.48

Table 5: Boundary Time of All Detectors (BT(1 day) represents the BT with respect to MAPE of 1 day)

Detector	2010	2011	2013	2023	2030	2033	2052
BT(1 day) /min	30	30	40	20	20	20	30
BT(1 week) /min	10	10	10	20	10	10	10
Detector	3034	3035	4004	4005	4050	4051	5062
BT(1 day) /min	30	20	40	30	120	70	30
BT(1 week) /min	10	10	10	20	60	60	10

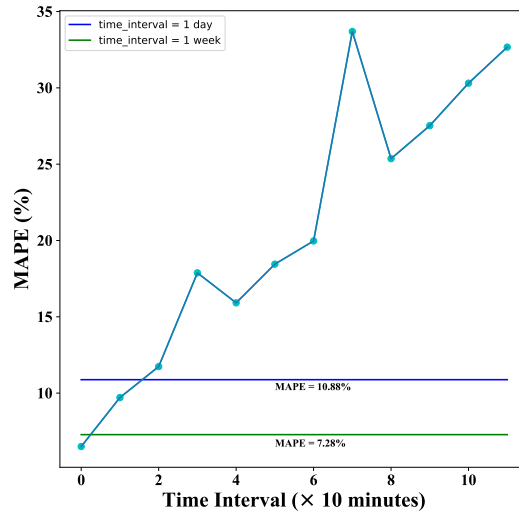


Figure 8: MAPE for Different Time Intervals

it works. Finally, we proposed our dynamic model DIDRN and demonstrated why it makes sense.

We presented the steps of processing data. We explained how raw data was processed into data that can be used in supervised learning and how we obtained the final predictions. Then, we performed several experiments and compared the performance of the different models. The results showed that DIDRN exhibited better performance than other commonly used models. From the view of MAPE, DIDRN obtained a maximum of 1.41% performance improvement compared to LSTM and DRN. From the view of RMSE and MAE, DIDRN achieved a maximum reduction of 9.

On the basis of these results, we analyzed our model by changing the time interval which is an input value and implemented the multi-step forward prediction which verifies the stability of our model. On the basis of performance at different time intervals, we demonstrated a recommended BT which could be a good choice in predicting traffic flow via the proposed model.

Despite the good performance, our model still exhibited some shortcomings. We only considered temporal pattern in this study. In future research, the spatial pattern would be considered and we will investigate spatial-temporal dependence in our model.

In addition, our model could perform traffic flow prediction well when the time interval or the period of our data was small. However, our model obtained poor performance when the time interval was slightly large. Future studies could be carried out to extend the model into a generalized version and obtain good performance when the time interval is both small and large, thereby indicating efficiency in both short-term and long-term traffic flow predictions.

References

- Baher Abdulhai, Himanshu Porwal, and Will Recker. Short term freeway traffic flow prediction using genetically-optimized time-delay-based neural networks. 1999.
- M. S. Ahmed and A. R. Cook. Analysis of freeway traffic time-series data by using box-jenkins techniques. *Transportation Research Record*, (no. 722):pp.1–9, 1979. URL <http://onlinepubs.trb.org/Onlinepubs/trr/1979/722/722-001.pdf>.
- Muhammad Tayyab Asif, Justin Dauwels, Chong Yang Goh, Ali Oran, Esmail Fathi, Muye Xu, Menoth Mohan Dhanya, Nikola Mitrovic, and Patrick Jaillet. Spatiotemporal patterns in large-scale traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):794–804, 2014. URL <https://ieeexplore.ieee.org/abstract/document/6678316/>.
- M.ASCE Billy M. Williams and F.ASCE Lester A. Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *J.Transport. Eng.*, 2003. URL <https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-947X%282003%29129%3A6%28664%29>.
- Enrique Castillo, José María Menéndez, and Santos Sánchez-Cambronero. Predicting traffic flow using bayesian networks. *Transportation Research Part B: Methodological*, 42(5):482–509, 2008. URL <https://www.sciencedirect.com/science/article/pii/S0191261507001300>.
- Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 32(5):361–378, 2017a. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12263>.
- Young-Jin Cha, Wooram Choi, Gahyun Suh, Sadegh Mahmoudkhani, and Oral Büyüköztürk. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 2017b. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12334>.
- Srinivasa Ravi Chandra and Haitham Al-Deek. Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, 13(2):53–72, 2009. URL <https://www.tandfonline.com/doi/abs/10.1080/15472450902858368>.
- Haibo Chen and Susan Grant-Muller. Use of sequential learning for short-term traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 9(5):319–336, 2001. URL <https://www.sciencedirect.com/science/article/pii/S0968090X00000395>.
- L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, Feb 2016. ISSN 1524-9050. doi: 10.1109/TITS.2015.2471812. URL <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7244203>.
- Weihong Chen, Jiyao An, Renfa Li, Li Fu, Guoqi Xie, Md Zakirul Alam Bhuiyan, and Keqin Li. A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial-temporal data features. *Future Generation Computer Systems*, 2018. URL <https://www.sciencedirect.com/science/article/pii/S0167739X18307398>.
- Tao Cheng, James Haworth, and Jiaqiu Wang. Spatio-temporal autocorrelation of road network data. *Journal of Geographical Systems*, 14(4):389–413, 2012.
- Steven Chien, Xiaobo Liu, and Kaan Ozbay. Predicting travel times for the south jersey real-time motorist information system. *Transportation Research Record: Journal of the Transportation Research Board*, (1855):32–40, 2003. URL <https://trrjournalonline.trb.org/doi/abs/10.3141/1855-04>.
- Steven I-Jy Chien and Chandra Mouly Kuchipudi. Dynamic travel time prediction with real-time and historic data. *Journal of transportation engineering*, 129(6):608–616, 2003. URL [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-947X\(2003\)129:6\(608\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2003)129:6(608)).
- Stephen D Clark, Mark S Dougherty, and Howard R Kirby. The use of neural networks and time series models for short

- term traffic forecasting: a comparative study. In *TRANSPORTATION PLANNING METHODS. PROCEEDINGS OF SEMINAR D HELD AT THE PTRC EUROPEAN TRANSPORT, HIGHWAYS AND PLANNING 21ST SUMMER ANNUAL MEETING (SEPTEMBER 13-17, 1993), UMIST. VOLUME P363*, 1993.
- G. A. Davis and N. L. Nihan. Nonparametric regression and short-term freeway traffic forecasting. *Journal of Transportation Engineering*, vol.117(no.2):pp.178–188, 1991. URL [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-947X\(1991\)117:2\(178\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(1991)117:2(178)).
- Stephen Dunne and Bidisha Ghosh. Regime-based short-term multivariate traffic condition forecasting algorithm. *Journal of Transportation Engineering*, 138(4):455–466, 2011.
- J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, vol.29(no.5):pp. 1189–1232, 2001. URL <http://www.jstor.org/stable/2699986>.
- Rui Fu, Zuo Zhang, and Li Li. Using lstm and gru neural network methods for traffic flow prediction. In *Chinese Association of Automation (YAC), Youth Academic Annual Conference of*, pages 324–328. IEEE, 2016. URL <https://ieeexplore.ieee.org/abstract/document/7804912/>.
- Qun Dai Gang Song. A novel double deep elms ensemble system for time series forecasting. *Knowledge-Based Systems*, 2017. URL <https://www.sciencedirect.com/science/article/pii/S0950705117303295>.
- Yuqing Gao and Khalid M Mosalam. Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, April 2018. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12363>.
- Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. Random process model for urban traffic flow using a wavelet-bayesian hierarchical technique. *Computer-Aided Civil and Infrastructure Engineering*, 25(8):613–624, 2010. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2010.00681.x>.
- Jianhua Guo, Wei Huang, and Billy M Williams. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C: Emerging Technologies*, 43: 50–64, 2014. URL <https://www.sciencedirect.com/science/article/pii/S0968090X14000382>.
- Hossein Hashemi and Khaled Abdelghany. End-to-end deep learning methodology for real-time traffic network management. *Computer-Aided Civil and Infrastructure Engineering*, May 2018. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12376>.
- Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360. IEEE, 2015.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. URL http://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.
- Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, 91:1, 1991. URL <https://link.springer.com/article/10.1007/2F03227308>.
- Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Trans. Intelligent Transportation Systems*, 15(5):2191–2201, 2014. URL <https://ieeexplore.ieee.org/abstract/document/6786503/>.
- Sheng Jin, Dian-hai Wang, Cheng Xu, and Dong-fang Ma. Short-term traffic safety forecasting using gaussian mixture model and kalman filter. *Journal of Zhejiang University SCIENCE A*, 14(4):231–243, Apr 2013. ISSN 1862-1775. doi: 10.1631/jzus.A1200218. URL <https://doi.org/10.1631/jzus.A1200218>.
- Matthew G Karlaftis and Eleni I Vlahogianni. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3):387–399, 2011. URL <https://www.sciencedirect.com/science/article/pii/S0968090X10001610>.
- Jan Koutník Bas R. Steunebrink Jurgen Schmidhuber Klaus Greff, Rupesh K. Srivastava. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28:pp. 2222 – 2232, 2017. URL <https://ieeexplore.ieee.org/document/7508408/>.
- Caglar Gulcehre Fethi Bougares Holger Schwenk Kyunghyun Cho, Bart van Merriënboer and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint*, arXiv:1406.1078, 2014. URL <http://arxiv.org/abs/1406.1078>.
- X. Su L. Li and Y. Zhang. Trend modeling for traffic time series analysis : An integrated study. *IEEE Transactions on Intelligent Transportation Systems*, vol. 16(no. 6):pp. 1–10, 2015. URL <https://ieeexplore.ieee.org/abstract/document/7180371/>.
- Tao Lei and Yu Zhang. Training rnns as fast as cnns. *arXiv preprint arXiv:1709.02755*, 2017. URL <https://arxiv.org/abs/1709.02755>.
- G. Leshem and Y. Ritov. Traffic flow prediction using adaboost algorithm with random forests as a weak learner. *International Journal of Mathematical, Computational, Physi-*

- cal, *Electrical and Computer Engineering*, vol. 1(no.1):pp.2–7, 2007. URL <https://pdfs.semanticscholar.org/d013/049dc55f011651cbc6c4ba27097be5e0bfad.pdf>.
- Ruimin Li and Huapu Lu. Combined neural network approach for short-term urban freeway traffic flow prediction. In *Advances in Neural Networks - ISNN 2009, 6th International Symposium on Neural Networks, ISNN 2009, Wuhan, China, May 26-29, 2009, Proceedings, Part III*, pages 1017–1025, 2009. doi: 10.1007/978-3-642-01513-7_112. URL https://doi.org/10.1007/978-3-642-01513-7_112.
- Ruimin Li and Geoffrey Rose. Incorporating uncertainty into short-term travel time predictions. *Transportation Research Part C: Emerging Technologies*, 19(6):1006–1018, 2011. URL <https://www.sciencedirect.com/science/article/pii/S0968090X11000842>.
- Ruoxing Li, Yachao Yuan, Wei Zhang, and Yali Yuan. Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Computer-Aided Civil and Infrastructure Engineering*, 2018. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12351>.
- Yi-zhou Lin, Zhen-hua Nie, and Hong-wei Ma. Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, 32(12):1025–1046, 2017. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12313>.
- Qingchao Liu, Bochen Wang, and Yuquan Zhu. Short-term traffic speed forecasting based on attention convolutional neural network for arterials. *Computer-Aided Civil and Infrastructure Engineering*, 33(11):999–1016, 2018.
- Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, Fei-Yue Wang, et al. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intelligent Transportation Systems*, 16(2):865–873, 2015. URL <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6894591>.
- Xiaolei Ma, Zhimin Tao, Yin Hai 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. URL <https://www.sciencedirect.com/science/article/pii/S0968090X15000935>.
- Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiya, and Hiroshi Omata. Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, June 2018. URL <https://onlinelibrary.wiley.com/doi/10.1111/mice.12387>.
- Mohammad Amin Nabian and Hadi Meidani. Deep learning for accelerated seismic reliability analysis of transportation networks. *Computer-Aided Civil and Infrastructure Engineering*, 33(6):443–458, 2018. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12359>.
- Vassilios Petridis, Athanasios Kehagias, Loukas Petrou, A Bakirtzis, S Kiartzis, H Panagiotou, and N Maslaris. A bayesian multiple models combination method for time series prediction. *Journal of intelligent and robotic systems*, 31(1-3):69–89, 2001. URL <https://link.springer.com/article/10.1023/A:1012061814242>.
- Jurgen Schmidhuber Sepp Hochreiter. Long short-term memory. *Neural Computation*, 1997. URL <https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735>.
- Brian L Smith, Billy M Williams, and R Keith Oswald. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4):303–321, 2002. URL <https://www.sciencedirect.com/science/article/pii/S0968090X02000098>.
- Xiang Song, Wenjing Li, Dongfang Ma, Dianhai Wang, Licheng Qu, and Yin Hai Wang. A match-then-predict method for daily traffic flow forecasting based on group method of data handling. *Computer-Aided Civil and Infrastructure Engineering*, June 2018. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12381>.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.
- Antony Stathopoulos, Loukas Dimitriou, and Theodore Tsikeris. Fuzzy modeling approach for combined forecasting of urban traffic flow. *Computer-Aided Civil and Infrastructure Engineering*, 23(7):521–535, 2008. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2008.00558.x>.
- Shiliang Sun, Changshui Zhang, and Guoqiang Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):124–132, March 2006a. ISSN 1524-9050. doi: 10.1109/TITS.2006.869623. URL https://www.researchgate.net/profile/Changshui_Zhang2/publication/3427947_A_Bayesian_Network_Approach_to_Traffic_Flow_Forecasting/links/55b1ea2208aed621ddfd6e22.pdf.
- Shiliang Sun, Changshui Zhang, and Guoqiang Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1): 124–132, 2006b. URL https://www.researchgate.net/profile/Changshui_Zhang2/publication/3427947_A_Bayesian_Network_Approach_to_Traffic_Flow_Forecasting/links/55b1ea2208aed621ddfd6e22.pdf.
- Huachun Tan, Guangdong Feng, Jianshuai Feng, Wuhong Wang, Yu-Jin Zhang, and Feng Li. A tensor-based method for missing traffic data completion. *Transportation Research Part C: Emerging Technologies*, 28:15–27, 2013. URL <https://www.sciencedirect.com/science/article/pii/S0968090X12001532>.

- Huachun Tan, Xuan Xuan, Yuankai Wu, Zhiyu Zhong, and Bin Ran. A comparison of traffic flow prediction methods based on dbn. In *CICTP 2016*, pages 273–283. ASCE, 2016. URL <https://ascelibrary.org/doi/abs/10.1061/9780784479896.026>.
- Zeren Tan and Ruimin Li. A dynamic model for traffic flow prediction using improved drn. *arXiv preprint arXiv:1805.00868*, 2018.
- Y. Tian and L. Pan. Predicting short-term traffic flow by long short-term memory recurrent neural network. *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, page pp. 153–158, 2015. URL <https://ieeexplore.ieee.org/abstract/document/7463717/>.
- JWC Van Lint. Online learning solutions for freeway travel time prediction. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):38–47, 2008. URL <https://ieeexplore.ieee.org/abstract/document/4459097/>.
- Eleni I Vlahogianni, John C Golias, and Matthew G Karlaftis. Short-term traffic forecasting: Overview of objectives and methods. *Transport reviews*, 24(5):533–557, 2004.
- Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Temporal evolution of short-term urban traffic flow: a nonlinear dynamics approach. *Computer-Aided Civil and Infrastructure Engineering*, 23(7):536–548, 2008. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2008.00554.x>.
- Jian Wang, Wei Deng, and Yuntao Guo. New bayesian combination method for short-term traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 43: 79–94, 2014. URL <https://www.sciencedirect.com/science/article/pii/S0968090X14000370>.
- Yibing Wang, Markos Papageorgiou, and Albert Messmer. Renaissance—a unified macroscopic model-based approach to real-time freeway network traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 14(3): 190–212, 2006. URL <https://www.sciencedirect.com/science/article/pii/S0968090X06000398>.
- Billy Williams. Multivariate vehicular traffic flow prediction: evaluation of arimax modeling. *Transportation Research Record: Journal of the Transportation Research Board*, (1776):194–200, 2001. URL <https://trrjournalonline.trb.org/doi/abs/10.3141/1776-25>.
- Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003. URL [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-947X\(2003\)129:6\(664\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2003)129:6(664)).
- Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281, 2004. URL <https://ieeexplore.ieee.org/abstract/document/1364002/>.
- Yuankai Wu, Huachun Tan, Yong Li, Feng Li, and Hongwen He. Robust tensor decomposition based on cauchy distribution and its applications. *Neurocomputing*, 223: 107–117, 2017. URL <https://www.sciencedirect.com/science/article/pii/S0925231216312218>.
- Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90:166–180, 2018. URL <https://www.sciencedirect.com/science/article/pii/S0968090X18302651>.
- Y. Zhang X. Jin and D. Yao. Simultaneously prediction of network traffic flow based on pca-svr. *Advances in Neural Networks ISNN*, vol. 4492(no. PART 2):pp. 1022–1031, 2007. URL https://link.springer.com/chapter/10.1007/978-3-540-72393-6_121.
- Jingxin Xia, Wei Huang, and Jianhua Guo. A clustering approach to online freeway traffic state identification using its data. *KSCE Journal of Civil Engineering*, 16(3):426–432, 2012.
- Qun Li Yue Huang Yingqi Li Xiaochuan Sun, Tao Li. Deep belief echo-state network and its application to time series prediction. *Knowledge-Based Systems*, 2017. URL <https://www.sciencedirect.com/science/article/pii/S0950705117302459>.
- Yilun Lin Xingyuan Dai, Rui Fu. Deeptrend: A deep hierarchical neural network for traffic flow prediction. *arXiv preprint arXiv:1707.03213*, 2017. URL <https://arxiv.org/abs/1707.03213>.
- W. Kang Z. Li Y. Lv, Y. Duan and F.-Y. Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, vol. 16 (no. 2):pp. 1–9, 2014. URL <https://ieeexplore.ieee.org/abstract/document/6894591/>.
- Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 316–324. ACM, 2011. URL <https://dl.acm.org/citation.cfm?id=2020462>.
- Allen Zhang, Kelvin CP Wang, Baoxian Li, Enhui Yang, Xianxing Dai, Yi Peng, Yue Fei, Yang Liu, Joshua Q Li, and Cheng Chen. Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network. *Computer-Aided Civil and Infrastructure Engineering*, 32(10):805–819, 2017. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12297>.
- Da Zhang and Mansur R Kabuka. Combining weather condition data to predict traffic flow: a gru-based deep learning approach. *IET Intelligent Transport Systems*, 2018. URL <http://digital-library.theiet.org/content/journals/10.1049/iet-its.2017.0313>.

Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017. URL <http://digital-library.theiet.org/content/journals/10.1049/iet-its.2016.0208>.

Weizhong Zheng, Der-Horng Lee, and Qixin Shi. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. *Journal of transportation engineering*, 132(2):114–121, 2006. URL [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-947X\(2006\)132:2\(114\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2006)132:2(114)).

A Algorithms

Algorithm 1: TimeSeriesToSupervised(R, lag)

```

1 // R: data returned by function
  Difference
2 // lag: lag is the number of time
  points used to predict data of next
  time point, in our research, lag = 1
3 shiftedData = shift R;
4 R.append(shiftedData);
5 return R;
```

Algorithm 2: Difference($D, interval$)

```

1 // D: raw data
2 // interval: time interval
3 Initialize R as  $\emptyset$ ;
4 for  $i$  from interval to D.length do
5   | D.append( $D[i] - D[i - interval]$ );
6 end
7 return R;
```

Algorithm 3: inverseDifference($historyData, \hat{y}, interval$)

```

1 // historyData: data used in Difference
2 //  $\hat{y}$ : the predicted difference of
  normalized traffic flow volume
  between two data points
3 // interval: time interval
4 Initialize D as  $\emptyset$ ;
5  $D = \hat{y} + history[-interval]$ ;
6 // history[-i] represents the last  $i^{th}$ 
  data point
7 return D;
```

Algorithm 4: scaleData($trainData, testData$)

```

1 // trainData: training data
2 // testData: test data
3  $Max_1 = trainData.max$ 
   $Min_1 = trainData.min$ ;
4  $Max_2 = testData.max$ 
   $Min_2 = testData.min$ ;
5 Initialize  $TD_1, TD_2$  as  $\emptyset$ ;
6 for each data point  $DP_1$  in trainData do
7   |  $newDP_1 = \frac{DP_1 - Min_1}{Max_1 - Min_1}$ 
     $TD_1.append(newDP_1)$ ;
8 end
9 for each data point  $DP_2$  in testData do
10  |  $newDP_2 = \frac{DP_2 - Min_2}{Max_2 - Min_2}$ 
     $TD_2.append(newDP_2)$ ;
11 end
12 return  $TD_1$  and  $TD_2$ ;
```

Algorithm 5: MainAlgorithm(D)

```

1 // D: raw data
2 Set all parameters;  $DV = Difference(D)$ ;
3  $SV = timeSeriesToSupervised(DV)$ ;
4 Split SV into training set  $TS_1$  and test set  $TS_2$ ;
5 Scale  $TS_1, TS_2$ ;
6 Use scaled  $TS_1$  to train the model;
7 Initialize the set of predict values PV as  $\emptyset$ ;
8 for each time point  $tp$  in  $TS_2$  do
9   | Use trained model to predict value at  $tp$ ;
10  | Add predicted value to PV;
11  | Add  $TS_2(tp)$  to  $TS_1$ ;
12  | Incrementally train model;
13 end
14 Calculate error rate;
```
