A Dynamic Model for Traffic Flow Prediction Using Updated Deep Residual Network

Zeren Tan[a] and Ruimin Li[b]

[a]Department of Civil Engineering, Tsinghua University, Beijing 100084, China
[b]Institute of Transportation Engineering, Tsinghua University, Beijing 100084, China*

May 2018

Abstract

Real-time traffic flow prediction can not only provide travelers with reliable traffic information so that it can save people's time, but also assist the traffic operation agency to manage traffic system. In this study, we build a dynamic prediction model with the improvement of Deep Residual Network (DRN). Based on traditional DRN model, we integrate the input and output of the i^{th} layer to the input of the $(i+1)^{th}$ layer and prove that each layer will fit a simpler function so that the error rate will be much smaller according to the hypothesis made by (He et al., 2016) while the model can still be trained fast. Then, we use the concept of online learning in our model to update pretrained model during prediction to rebuild the model with time going on. The result shows that our model has higher accuracy than some state-of-the-art models when they are applied in the traffic volume prediction of the 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, quickly and timely traffic flow prediction is one of the major tasks of intelligent transportation systems (ITSs)(Lu et al., 2015; Xingyuan Dai, 2017). It is of practical significance for individuals, companies and governments to make decisions according to real-time and predicted traffic flow(Chen and Englund, 2016). Office workers need this kind of information to schedule the trip to their work place. Traffic operation agency may use the predicted traffic flow information to plan vehicle lanes(Yuan et al., 2011; Wu et al., 2018). However, despite the long-lasting research and developments worldwide, accurate and short-term traffic flow prediction remains challenging to researchers for decades because of its stochastic and nonlinear characteristics. At the beginning of traffic flow prediction research, researchers mainly used linear methods, such as autoregressive integrated moving average (ARIMA) (Ahmed and Cook, 1979). For their simplicity and convenience, these linear models have been improved continuously and applied in traffic flow prediction, for example, multi-variable linear regression (MVLR) (L. Li and Zhang, 2015). Despite improvement, there is still some room to improve the prediction accuracy, for the nonlinear nature of traffic flow. Therefore, it is still of great significance for researchers to find more powerful methods to predict traffic flow more precisely.

From about 40 years ago, some machine learning algorithms have showed good performance in many tasks such as prediction and classification, so researchers have begun to use machine learning algorithms like support vector regression (SVR) (X. Jin and Yao, 2007) and k-Nearest Neighbor (k-NN) (Davis and Nihan, 1991) in traffic flow prediction recently. In spite of performance increase, these approaches cannot consider the entire characteristics in traffic flow and do not have satisfactory performances for the complexity of traffic flow. Thus, traffic flow prediction are still one of the research hotspots and are favored by researchers from civil engineering, automation, industrial engineering as well as computer science.

In recent years, deep learning methods, which are a subset of machine learning methods, have shown great performance in many tasks in researches related to computer science, such as image classification, machine translation and speech recognition. More and more researchers who are working on other fields, for example, researchers from civil engineering are inspired by deep learning and begin to apply these 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), reliability of transportation network(Nabian and Meidani, 2018), traffic speed estimation(Liu et al., 2018) as well as real-time transportation management (Hashemi and Abdelghany, 2018). Recently, with the development of deep learning, traffic flow prediction has become an appealing field for applying Deep Learning approaches and models, such as Recurrent Neural Network (RNN) (Kyunghyun Cho and Bengio, 2014), Long Short-Term Memory network (LSTM) (Tian and Pan, 2015), Gated Recurrent Unit (GRU) (Fu et al., 2016), Stacked Auto-Encoders (SAEs)

*Email:lrmin@tsinghua.edu.cn

(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).. In general, these models have complex network architectures that can capture nonlinearities in traffic flow. Hence, these previous studies showed that they perform better in forecasting traffic flow than the chosen 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).

Among all these deep learning models, RNN (Kyunghyun Cho and Bengio, 2014), LSTM (Tian and Pan, 2015) and GRU (Fu et al., 2016) are three of the most commonly used models. However, the architectures of these models are so complicated that it takes a great amount of time to train them before thetraining process converges. Since it is hard to train, people often have to stack less layers and set the training epochs to a small number, which will result in lower accuracy to some extent, or people may use some techniques such as dropout to reduce the size of training set and test set which does not allow people to get a higher accuracy when the data set is not large enough. Therefore, it is difficult to use these models in practical applications. On the other hand, previous studies were mainly based on static models, that is, the model will not be updated once it was trained. Nevertheless, realistic traffic flow condition is the continuous updated includes significant uncertainty. In addition, traffic flow is periodic data with one day and one week as its periods(Tan et al., 2013, 2016; Wu et al., 2017). In addition, traffic flow includes significant uncertainty and is changing all the time(Chen et al., 2018). Therefore, we need to build the prediction mechanism of real-time updating in order to adapt to the continuously changing traffic flow status.

In this paper, 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 some real data are available, which is more powerful in practical application.

The rest of this paper is organized as follows. Section 2 reviews the existing works on short-term traffic flow prediction. Section 3.1 demonstrates basic ideas of DRN. Section 3.2 and Section 3.3 explain how we further update DRN and how we design the dynamic model, respectively. Section 4 develops experiments and compares our model's performance with several popular models as well as analyze the performance of our model in detail. Finally, section 5 concludes this paper and discusses future works. Appendix lists all algorithms used in this research.

2 Literature Review

Since traffic flow prediction is one of the main tasks of ITSs, researches on traffic flow prediction have been ongoing for many years up to now. New models are constantly being proposed while the performance of proposed models increases in the mean time. Generally, existing numerous traffic flow prediction methods can be divided into two main categories: parametric methods and 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) were often used by researchers. These approaches require predetermined model architecture and that the parameters of the model are calculated by empirical data (Xingyuan Dai, 2017). Some improved models based on ARIMA like 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 explicit architecture but require a huge amount of data and that the traffic condition is in a stationary process. Hence, it may be impossible to use these approaches when sufficient data are not available and its performance would be unsatisfactory when the detected traffic flow of a certain location is not stable enough. In addition, parametric models requires prior knowledge to assume some finite set of parameters. Thus, the setting of these parameters would be the thing of trick. In most cases, traffic flow data have characteristics of randomness and there is uncertainty in traffic flow data. Therefore, those parametric approaches cannot perform very well in most cases. They are limited both theoretically and practically.

For nonparametric models, the structure and parameters need not to be fixed in advance(Ma et al., 2015). Around the beginning of 21st century, researchers began to pay more attention to 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 these methods may not be 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) to traffic flow prediction. However, BCM does not pay attention to the relevance between historic traffic flows and traffic flow at current time point (Wang et al., 2014). Therefore, this model would not be the best choice when it comes to traffic flow prediction since traffic flow is sequential data which means current data are related to historical data. On the other hand, due to the excellent performances of Artificial Neural Networks (ANNs) 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), researchers from many fields, including transportation indeed, have paid more and more attention to them since the birth of ANNs.

Recently, with the rapid development of deep learning, a lot of deep learning models have been proposed by computer scientists. Deep learning models can represent the complex features of traffic flow without prior hypothesis (Lv et al., 2015; Chen et al., 2018). The thing you need to do is develop a neural network model by adding layers and feed your data into the network. Your do not need to know what has happened in the network while training and predicting thoroughly. But it has been proven theoretically by Hornik et al. (1989) that one layer neural network is able to represent any Borel measurable function. 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 to traffic flow prediction and showed superior performances compared to traditional models. Y. Lv and Wang (2014) proposed a greedy layer-wise unsupervised learning algorithm to train the model so that the training time can be reduced significantly without loss of accuracy. Tian and Pan (2015) proposed a model called LSTM RNN for short-term traffic flow prediction and showed their model can achieve higher accuracy than classic models such as SVM. Fu et al. (2016) proposed GRU NN for traffic flow prediction and showed that GRU and LSTM outperform ARIMA. Chen et al. (2018) developed a fuzzy deep learning model called FDCN and their results showed that the model is more powerful in representing capability. Their model is stable in terms of RMSE when doing mmulti-step forecasting.

Sequential data are correlated with data in the context. Nevertheless, common neural networks cannot capture this correlation among data. In the 1990s, RNN (Hochreiter, 1991) was developed to solve this problem. However, although RNN can capture correlation among data, it will soon forget this correlation, that is, it does not have a long-term memory (Zhao et al., 2017; Ma et al., 2015). Hence, when the data are strongly correlated and historical data have a great influence on consequent data long time later, it is not wise to choose RNN as training model. In 1997, long short-term memory (LSTM) networks (Sepp Hochreiter, 1997) were proposed and set accuracy records in various applications domains. LSTM can remember correlation for a long time and thus has an outstanding performance in sequential modeling. However, it has been proven that both RNN and LSTM are very hard to train since the architecture of these models are too complex (Lei and Zhang, 2017). It often takes researchers hours or even days to train these models so training model is time-consuming especially when the the dataset is very large. Although they have good performance, to use them in practice is sometimes impossible. In order 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 unacceptable. In addition, Klaus Greff (2017) made a good comparison of the variants of LSTM under a large-scale study on many different datasets and found that they are all the same and none of its modifications improve performance significantly.

Besides RNN and LSTM, researchers have also tried some other deep learning models. Gang Song (2017) proposed a novel double deep ELMs(Extreme Learning Machines) ensemble system (DD-ELMs-ES) which uses three deep ELM models as basic models and their model demonstrated better generalization performance than some state-of-art algorithms. However, their model only focuses on one-step forecasting. Xingyuan Dai (2017) implemented DeepTrend which decomposes original traffic flow data into trend and residual components. They demonstrate that DeepTrend can noticeably improve the 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. Experiments results demonstrate that DBEN has a good performance in learning speed and short-term memory capacity. However, there are a large number of parameters in their model so parameter tuning would be a really hard work.

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, when networks become deeper, it will be difficult to train them and the problem of gradient vanishing will arise. Their model is easy to train and can solve the problem of gradient vanishing effectively even if the deep neural network is several times deeper than other networks. For a normal deep neural network, the maximum depth that people have ever used is less than 100. But for DRN, a network with several hundred layers can be easily created and train it in a even shorter time. Thus, we are inspired by the ideas behind DRN and attempt to apply it in traffic flow prediction. In order to apply it in our field, we improve the architecture of DRN and make it more powerful. Our result shown in Section 4 illustrates this point.

3 Model Development

3.1 Original DRN

When deep learning began to boost, people simply stacked layers and expected that deeper is better. They give the network an input x and let the intermediate layers fit a map H to get 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 get a higher accuracy. However, it turns out that when the depth of the network increases, both training and test accuracy get saturated, that is, degradation arises(He et al., 2016). Figure 1 shows the architecture of simply stacked neural network.

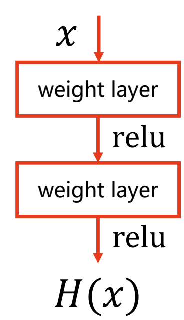


Figure 1: Architecture of common neural network

DRN is proposed to solve the problem of degradation. Instead of hoping each stacked layer fits a desired mapping directly. He et al. (2016) let some layers fit a residual mapping.

Concretely, assuming that the network is expected to fit $H(x)$, He et al. (2016) let the sacked layer fit another mapping $F(x) =$

$H(x) - x$. Then the desired mapping $H(x)$ can be replaced by $F(x) + x$.

They hypothesize that it is easier to fit the residual mapping $F(x)$ 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.

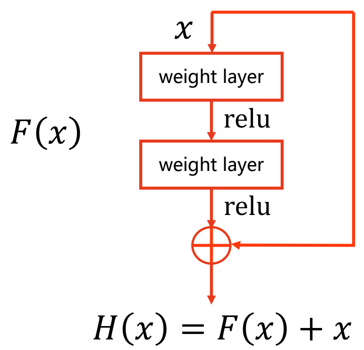


Figure 2: Architecture of residual neural network

This simple change makes a lot of difference. The network becomes much easier to train. Deep residual network has helped Kaiming and his team win 1st prize in the ILSVRC classification competition and many other competitions. Before DRN existed, the winner network of ILSVRC had at most 32 layers while when DRN participated in this competition, it had 152 layers. What's more, the accuracy had a significant increase.

3.2 Updated DRN

To keep more information when information is transferred among layers and thus get higher accuracy, we further improve the architecture of DRN and use it in traffic flow prediction. Figure 3 shows architecture of our improved network.

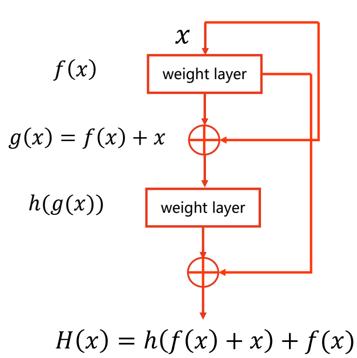


Figure 3: Our improved residual network

The network works as follows:

- Denote the input as x, and the first layer fits the function $f(x)$.
- Then we add these two outputs together and we get $g(x) = f(x) + x$.
- The second layer takes $g(x)$ as input and maps it into $h(g(x))$.
- Similarly, we then add the output of the first layer $f(x)$ and the output of the second layer $h(g(x))$ and we get $H(x) = h(f(x) + x) + f(x)$.

From the aforementioned steps, it is clear that the task of fitting a complex function $H(x)$ is divided into fitting two simpler mappings f and h, respectively. Based on the hypothesis of He et al. (2016), it will make the layers work more efficiently since it is easier to fit these two simple mappings. Consequently, it can reduce the error rate of fitting these two mappings and eventually reduce the overall error rate.

3.3 Dynamic Model

In practice, traffic operation center will not use this model in the exact place that our training data come from, that is, our model needs to be transferred easily among different places. Besides this, traffic flow on a specific road may change over time. In other words, the traffic flow after one month or one year may have significant difference with the current one, especially in e rapidly developing cities. In the DRN model, the weights and parameters were obtained by using the used training data. These weights and parameters can fit the used data very well but it may not fit a new data well, that is the previous model may not have good performances on new data. Therefore, it is unreasonable to simply use the old model with only one training process to predict traffic flow without updating it. It is also not surprising that we cannot expect this old model to have good performances when the traffic flow has changed significantly. Hence, traffic flow prediction models should be updated constantly. Based on the ideas stated above, we choose to design a dynamic model based on improved DRN.

In order to update pre-trained model, we use the basic idea of incremental learning to implement a dynamic model.

The complete process is as follows:

Step 1 We use the collected data to pre-train a basic model.

Step 2 Then we use our model to do prediction work.

Step 3 After some time,the basic mode finishing some steps of prediction, and we get some new real data.

Step 4 We combine these new data into a new training set and use this new training set to train our trained model again.

Step 5 Repeat Step 2-Step 4.

Figure 4 shows the process of these steps.

With these steps, the model will be continuously updated through absorbing new data. Therefore, this model will fit the practical conditions better and give more accurate predictions.

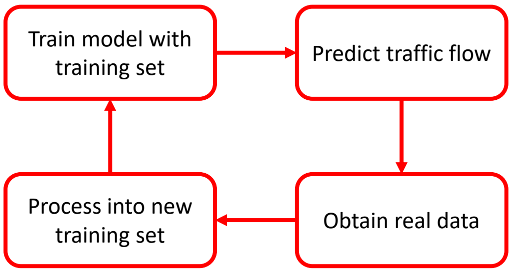


Figure 4: Dynamic Model

3.4 Model Implementation

In this section, we will talk about the whole process of implementing the model.

The process is divided into several steps:

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

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

It is reasonable to assume that there are some trend characteristics in traffic flow data. So we subtract adjacent data to get the difference vector $D = [D_1, D_2, \cdots, D_{n-1}]$. Where

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

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

First, the time-step should be chose. Time-step represents how many previous data points we use 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

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

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

Then we get the entire data set.

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

Step 3 In order to accelerate the speed of training and prediction, a simple approach that people often use when developing deep learning model and doing experiments is normalization. Since some elements in D are positive and others are negative, X and Y are scaled into $(-1, 1)$. We define the resulted vectors as $scaled\_X$ and $scaled\_Y$. They are defined as follows:

$$scaled\_X = \begin{bmatrix} scaled\_X_1 \\ scaled\_X_2 \\ \vdots \\ scaled\_X_{n-1} \end{bmatrix} \quad scaled\_Y = \begin{bmatrix} scaled\_Y_1 \\ scaled\_Y_2 \\ \vdots \\ scaled\_Y_{n-1} \end{bmatrix}$$

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

$$scaled\_X_i \in (-1, 1), i = 1, 2, \cdots, n-1$$

$$scaled\_Y_i \in (-1, 1), i = 1, 2, \cdots, n-1$$

Then the data set becomes

$$\begin{bmatrix} scaled\_X_1 & scaled\_Y_1 \\ scaled\_X_2 & scaled\_Y_2 \\ \vdots & \vdots \\ scaled\_X_{n-1} & scaled\_Y_{n-1} \end{bmatrix}$$

Step 4 When predicting, we invert Step 1 and Step 3 to get the final prediction data.

All algorithms (Algorithm 2 $\sim$ Algorithm 5) used to implement the whole process are presented in Appendix A

4 Experiments and Results

4.1 Data Source

Our data had been chosen from the traffic detectors located on the ring roads in Beijing, China. Basic traffic flow data are collected by 14 microwave detectors. They are numbered as 2010, 2011, 2013, 2023, 2030, 2033, 2052, 3034, 3035, 4004, 4005, 4050, 4051, 5062. The interval of time of these traffic flow data is 10 minutes. The time span is 61 days from June 1^{st}, 2013 to July 31^{st},2013. Therefore, there are 122976 data points. 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.

All data processed during the course of this study and the code for generating the neural network are subject to licensing and can be made available upon request to the authors.

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 are defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}-y)^2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y_i}-y_i|}{y_i} \times 100\%$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}-y|$$

where y_i is the real value and $\hat{y_i}$ is the forecasted value.

RMSE and MAE represent the deviation from the predicted values and the detected values. They will expand as the range of the data is expanded. Therefore, we cannot evaluate our model just on the basis of the absolute value of RMSE and MAE. The range of our data should be considered at the same time. However, MAPE is a relative measure of the deviation from the predicted values and the ground truth values. It is a percentage error so it is not related to the range of our data. We use these three performance indexes to evaluate two aspects of the selected models: 1) absolute error 2) relative error.

4.3 Performance Analysis

In order to analyze the efficiency of our model, the following experiments have been carried out.

(a) The performance of DIDRN in terms of different number of layers has been inspected to see how the prediction accuracy changes when the number of layers changes as shown in Section 4.3.1.

(b) The performance of different models on short-term traffic flow prediction as shown in Section 4.3.2.

(c) The performance of DIDRN under different prediction time intervals as shown in Section 4.3.3. Concretely, we increase time interval gradually from 10 minutes to 7 days in order to evaluate the stability of our model, analyze its performance and figure out which is better for use when predicting traffic flow.

All the experiments are run on a laptop with @2.60 GHz processor, 8.0 GB RAM and NVIDIA GeForce GTX 960M. All models were coded in Python 3.6 with Keras and Tensorflow framework and they were compiled by Anaconda Jupyter Notebook.

4.3.1 Impact of the Number of Layers

In the first place, the input dimension should be determined, that is the number of data points we input to the models as one input point. Foe example, if the input dimension is chosen as 5, then the models use 5 data points to predict the traffic flow of next time point. After parameter tuning, the input dimension is chosen as 1.

As mentioned above when network becomes deeper, accuracy rate will firstly increase and then decrease. To verify this, experiments on DIDRN with different number of layers have been carried out. In this experiments, time interval is chosen as 10 minutes. Figure 5 shows the result. As it is revealed, when the network is shallow, it has a fairly good performance, but when the network becomes deeper and deeper, the error rate rises simultaneously. When the network has over 40 layers, the error rate increases significantly. We choose more small numbers especially numbers approximate 16 and find that when the number equals to 16, the error rate is the lowest among all numbers experimented. Although when the number is 36, it still has a similarly error rate, the training time becomes longer as the network becomes deeper. When the number of layer exceeds 40, the error rate has a clear trend of increasing and it is much higher than the network is shallower so we choose less points in our experiments. Our result thoroughly verifies the problem reported in (Srivastava et al., 2015; He and Sun, 2015; He et al., 2016).

4.3.2 Short-term prediction

In order to evaluate the efficiency of our proposed model, we develop several experiments on our model and several similar models, including one layer LSTM, deep LSTM, and DRN. From the result in Section 4.3.1, except for one layer LSTM, the other three models have 16 layers. In order to change the input dimension in to the output dimension, we add some layers in both DRN and DIDRN but they still have 16 layers in total.

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

Table 1 and Figure 6 show the performance.

Through Table 1, it can be found that DIDRN has more outstanding performance than other models. One layer LSTM has a much worse performance than other models since it is shallower. Deep LSTM seems to have a satisfying performance, however, on our machine, it actually takes us nearly 5 minutes to process data from one detector and train it before it converges. DRN shares a similar but lower error rate with deep LSTM and it is much easier to train. It only takes us about 50 seconds to train DRN. On the other hand, although DIDRN is derived from DRN, it has a lower error rate than DRN. This implies our model is efficient enough. In addition, it is not hard to train. We can finish training it within 100 seconds which is a little longer than that of DRN but it is much shorter than that of deep LSTM. The reason why performances of DRN and Deep LSTM are similar is that when the network is not deep enough, they are comparably accurate(He et al., 2016), just similar with the result illustrated by their paper. When the network becomes really deep, their variation would be revealed. But generally, DRN needs less time to be trained.

From Table 1, it can observed that model accuracy of dataset 4050 is much lower than othe datasets. We plot the data points of dataset 2010 and dataset 4050 as shown in Fig 4.3.2. From the figures, the conclusion can be drawn that data of dataset 4050 is not as stable as other dataset and it is more discrete. Models evalued generally cannot perform very well when this kind of circumustence occurs.

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**

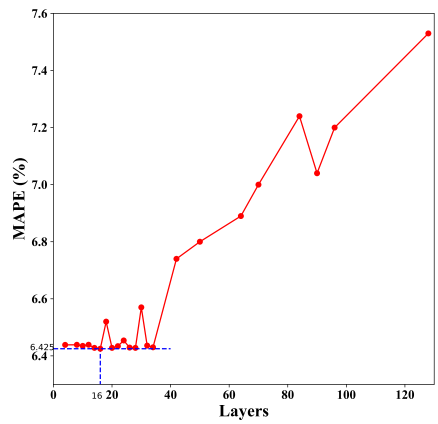
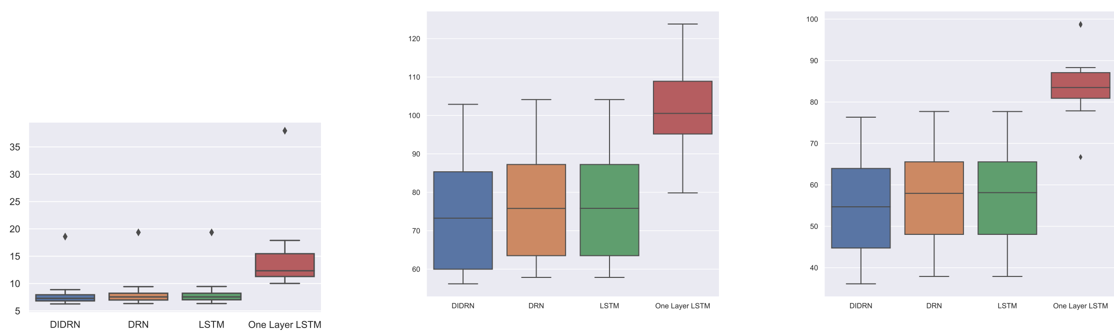


Figure 5: Impact of number of layers



(a) MAPE (b) RMSE (c) MAE

Figure 6: Forecast Error of Different Models

Figure 7 shows the a comparison of predicted data of DIDRN and measured data on dataset 2010.

4.3.3 Further Analysis on Performance of DIDRN

Under the following motivations, this set of experiments have been conducted.

(1) Our model shows good performance when the time interval is chosen as 10 minutes, but commonly, a model will have a worse performance when time interval becomes larger. We are motivated to thoroughly inspecting the impact of time interval on model performance.

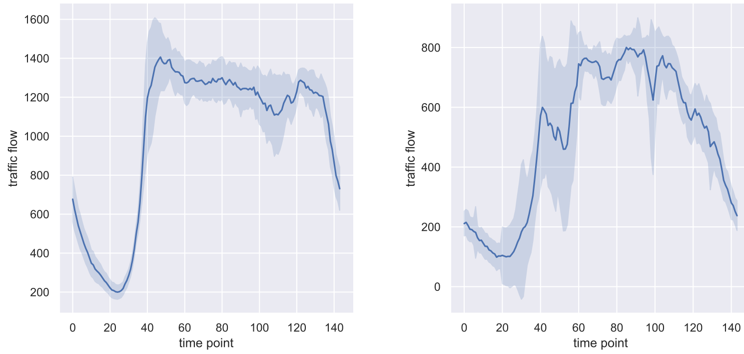
(2) Table 1 shows that when time interval is 10 minutes, the MAPE is good. But if there exists another time interval that our model has a even better performance? We want to experimentally make this explicit in this section.

(3) The practical scenario is complicated, we cannot always get the data we need. Therefore, a problem arises, if our model has the best performance when time interval is 10 minutes, but we do not have this kind of data, that is, we do not have historical data 10 minutes ago when predicting traffic flow, then what should we do? If we have data 1 day ago or 1 week ago, can we use these data? How about the accuracy? This is third motivation of presenting this section.

In the following discussion, our analysis is conducted mainly based on MAPE because it is simple and intuitive. The data of detector 2010 are chosen to demonstrate the results by random selection but data from either detector would show similar results.

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

From Table 2, we can see that while the time interval becomes larger from 10 minutes to 1 hour and 2 hours, the accuracy decreases monotonously. This is congruent with our intuition. When the time interval becomes larger, we need 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, it is obvious that the accuracy would decrease.



(a) Dataset 2010

(b) Dataset 4050

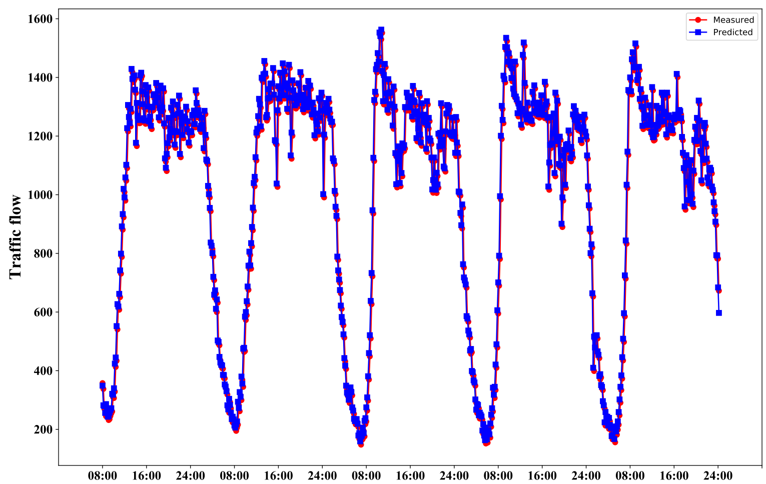


Figure 7: Measured data and predicted data

However, when we use the traffic flow of the previous day to predict that of the next day, the accuracy increase significantly to 10.88% in terms of MAPE. This is not strange. On one hand, since traffic flow data are stable and have periodic nature(Tan et al., 2013, 2016; Wu et al., 2017), traffic flow of the same time in the previous day is quite similar to that of the predicted day. On the other hand, traffic flows 1 hour or 2 hours later can be really different with that of the current time point. For example, traffic flow at 7:00 a.m. has significant change compared to traffic flow at 8:00 a.m. and 9:00 a.m. since they are peak hours. People go to work during peak hours and consequently, lead to a sudden increase in traffic flow. Because the only input is data of 7:00 when the model attempts to predict traffic flow at 8:00 or 9:00, the model do not have enough information to predict this sudden increase so the predicted values have a larger deviation from the detected values.

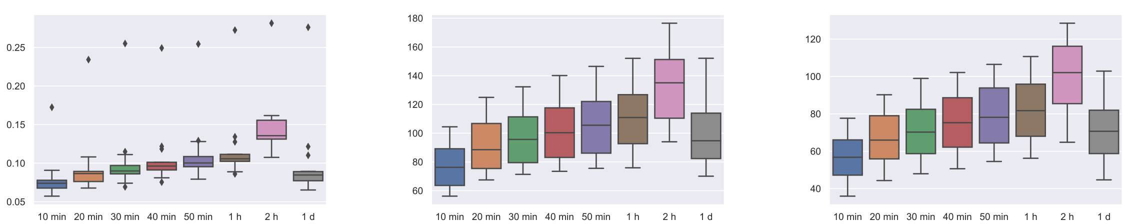
Table 3 show the MAPE of different detectors in terms of one day forward prediction, that is, we use the data of the previous days to predict the traffic flow of the next day. As it is revealed, the model have a stable performance on data from different detectors. Though the model has a much poor performance on data from detector number 4050, it is simply because the data of this detector are not good enough. That is, the traffic flow of the place this detector is located is not stable over time. We cannot expect our model to have good performance when the data is not good and stable enough. This observation is supported by results shown in Table 1 too. As presented in Table 1, the MAPE of this dataset, namely data from detector number 4050, is much higher than other datasets since each of the four models has an error rate at this dataset that is as literally twice as the highest error rate it gets from other datasets.

In order to further analyze the impact of time interval, we gradually increase the time interval from 1 day to 7 days.

Table 4 shows the result. It can be observed that when the time interval is ranging from 2 days to 6 days, the error rate ranges from 13.96% to 10.45%. This is because when time interval be-

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: 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%

comes larger, the predicted traffic flow has less relation with the input traffic flow. The model has less information to update its weights in order to perform well both in training set and test set.

There is a large increase between 6 days and 7 days. Due to symmetry, the error rate of 6 days should be similar with the error rate of 1 day and it can be observed from our result too. Since traffic flow are a kind of periodic data and the period is exact 7 days, it is not surprising that the flow one week later is similar with the current flow. Due to aforementioned property, our model can perform better and thus has a higher accuracy.

We increase the time interval from 10 minutes to 2 hours and compare the performance with the result we have already presented. The result of time interval less than 2 hours and result of 1 day and 7 days are show in Figure 8. In Figure 8, it is clear that MAPE increases when time interval increases. The performance given by our model is best when time interval equals to 10 minutes. Performance of 1 day is better than that of 30 minutes but is worse than 20 minutes while performance of 1 week is better than 20 minutes but is worse than 10 minutes. We refer this 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, as boundary time (BT). The experiment is also conducted on data from all detectors. Table 5 lists the boundary time for all of them.

According to the revealed result, the following conclusion could be drawn:

1. If data of the previous time point are available, it is the best choice that using those data to predict traffic flow of next time point. That is, time interval should be chosen as 10 minutes.
2. If data of the previous week are available or the task is doing long-term prediction, it is better to use these data.
3. Generally, when time interval is greater than 30 minutes, the performance would be better if time interval is chosen as 1 day or 1 week. In practical application, the boundary time can be obtained according to history data and it could be a reference for choosing time interval when predicting traffic flow.

5 Conclusion and Future work

In this paper, we demonstrate the basic ideas of deep residual network. It turns out that DRN is much simpler to train and have an excellent performance. Then we explain how we are inspired by DRN and and how we improve it to do our research. We show the architecture of our network and how it works. After that, we propose our dynamic model DIDRN and demonstrate why it makes sense.

We show the entire process of processing data step by step. We explain how we process the raw data into data that can be used in supervised learning and how we get the final predictions. Next, we develop several experiments and compare performance of different models. It turns out that DIDRN has better performance than some popular models. From the view of MAPE, DIDRN has a 1.41% performance improvement at most comparing to LSTM and DRN. For RMSE and MAE, DIDRN can have a mostly 9 of reduction.

Upon the result, we analyze our model by changing the time interval which is an input value and implementing multi-step forward prediction which shows the stability of our model. Based on performance of different time intervals, we show a recommended time boundary which could be a good choice when using the model to predict traffic flow.

To summarize, our main contributions are as follows:

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

Despite the good performance, our model still have some shortcomings. We only take temporal pattern into account in this paper. In future work, the spatial pattern would be considered and we will make our model learn spatial-temporal dependence.

In addition, our model can do traffic flow prediction well when the time interval is short or is the period of our data. But it has a poor performance when the time interval is a little larger. Future work can be done to extend the model to a more generalized version so that the model can perform well when time interval is both short and large. That is, the model is efficient in both short-term and long-term traffic flow prediction.

References

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

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

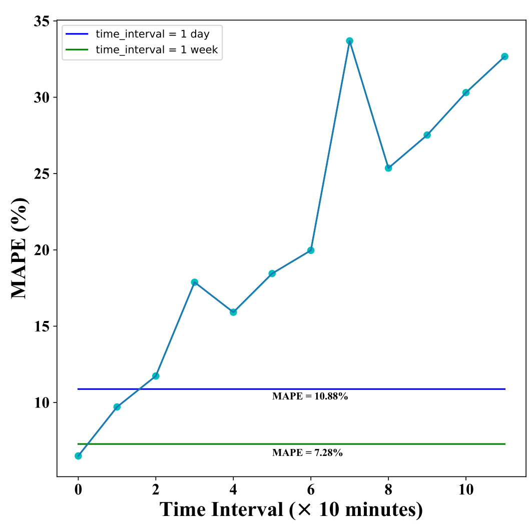


Figure 8: MAPE for Different Time Intervals

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.

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).

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).

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. *Knowle dge-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%2FBF03227308.

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 Koutnık 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 Merrienboer 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, Physical, 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.

Hua-pu Lu, Zhi-yuan Sun, Wen-cong Qu, and Ling Wang. Real-time corrected traffic correlation model for traffic flow forecasting. *Mathematical Problems in Engineering*, 2015, 2015. URL https://www.hindawi.com/journals/mpe/2015/348036/.

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, 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. URL https://www.sciencedirect.com/science/article/pii/S0968090X15000935.

Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiyama, 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 Yinhai 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 Tsekeris. 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, 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).

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%2F978-3-540-72393-6_121.

Qun Li Yue Huang Yingqi Li Xiaochuan Sun, Tao Li. Deep belief echo-state network and its application to time series prediction. *Knowle dge-Base d 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).

A Algorithms

Algorithm 1: TimeSeriesToSupervised(R, lag)

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

Algorithm 2: Difference($D, interval$)

```
// D: raw data
// interval: time interval
Initialize R as ∅;
for i from interval to D.length do
    D.append(D[i] − D[i − interval]);
end
return R;
```

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

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

Algorithm 4: scaleData($trainData, testData$)

// $trainData$: training data
// $testData$: test data
$Max_1 = trainData.max \quad Min_1 = trainData.min$;
$Max_2 = testData.max \quad Min_2 = testData.min$;
Initialize TD_1, TD_2 as $\emptyset$;
for *each data point* DP_1 *in* $trainData$ **do**
  $newDP_1 = \frac{DP_1 - Min_1}{Max_1 - Min_1}$ TD_1.append($newDP_1$);
end
for *each data point* DP_2 *in* $testData$ **do**
  $newDP_2 = \frac{DP_2 - Min_2}{Max_2 - Min_2}$ TD_2.append($newDP_2$);
end
return TD_1 and TD_2;

Algorithm 5: MainAlgorithm(D)

// D: raw data
Set all parameters; DV = `Difference(D)`;
SV = `timeSeriesToSupervised`(DV);
Split SV into training set TS_1 and test set TS_2;
Scale TS_1, TS_2;
Use scaled TS_1 to train the model;
Initialize the set of predict values PV as $\emptyset$;
for *each time point tp in* TS_2 **do**
  Use trained model to predict value at tp;
  Add predicted value to PV;
  Add $TS_2(tp)$ to TS_1;
  Incrementally train model;
end
Calculate error rate;