A Short-Term Traffic Flow Prediction Based on Recurrent Neural Networks for Road Transportation Control in ITS

[1]Wafa Shafqat, Sehrish Malik, Yung-cheol Byun, Do-Hyeun Kim

Abstract: Background/Objectives: In order to overcome the rising issue of traffic congestion, effective and exact traffic flow information is needed. Though, there have been lots of research and work being done on traffic predictions; still this field of research needs more attention. Methods/Statistical analysis: In this work, we have collected some real time traffic data for analyzing different flow patterns under different environmental conditions. In this paper, we present a short-term traffic flow prediction using RNN (Recurrent Neural Networks) for Road Transportation Control in ITS. Findings: Prediction of accurate traffic rate flow at any given time interval which is of vital importance in assisting and managing the road traffic conditions in smart cities. Improvements/Applications: Applied system uses deep learning technique for accurate predictions of traffic flow rate at any specific time.

Keywords: Traffic Flow Prediction, Recurrent Neural Networks (RNNs), Road Transportation, Long Short Term Memory (LSTM), Flow rate.

I. INTRODUCTION

Through the prompt evolution of automobiles and the development of smart cities, the yearly rate of traffic congestions in cities is increasing quickly, as a result the poor competence of transportation networks is observed, and wastage of time, fuel and air pollution are the byproducts of it. Thus, research on the prediction of traffic flow rate is critical as well as it has been considered as a fundamental issue of intelligent transport management [1], which is moreover an imperative resource to conduct the logical decision-making of traffic controlling. Primary analysis of jamming incidence and forecast traffic flow progress are measured to be a key element to define traffic bottlenecks.

In the modern era, enormous data availability has become a crucial matter in every field and industry. The availability of such abundant data raises question of how to give useful meaning to this data in an effective and efficient way.

The increasing traffic congestion is a crucial problem that is consistently being faced by the Intelligent Transportation Systems (ITS). According to a report [1], the number of vehicles will be doubled by 2040 globally. With this increased number of vehicles every day, traffic congestion mainly in large cities is taking an inescapable status primarily in peak hours. However, with the advancements in Artificial Intelligence (AI) techniques, we can overcome this issue by analyzing and learning the traffic flow patterns and predicting traffic status at any given point and time. There have been many experiments and studies performed to predict traffic flows by using different methods. We can find architectures or models like historical averages, Feed Forward Neural Networks, deep neural networks and recurrent neural networks being applied to traffic data for flow predictions [2-5]. In this paper, we also aim to predict traffic flow rate by applying some deep learning techniques. We have used RNNs to train the model. We did some preprocessing on the input data to get it into the desired form. Data was aggregated based on time intervals e.g., time interval of 15 minutes etc. The proposed model will take an input traffic flow data and predict the traffic flow rate in future i.e., number of vehicles at a given time.

Over the previous years, various studies focused on the detection of traffic congestion and presented the analysis and prediction of traffic flow [2-5]. Though, many of these studies are based on simulations and mathematical derivations to most of these studies rely on mathematical equations or simulation techniques to define the development of network congestion. As, it's not just about vehicles, a transportation network is heavily influenced by living beings, weather conditions, accidents, ignorance towards traffic rules etc., and many other factors, therefore it is very challenging to use mathematical models to represent them precisely. These traditional methods include several learning algorithms, e.g., ANNs (artificial neural networks).

This era being an era of IoT, there are many technologies that are constantly acquiring traffic data through different sensors, introduced us with an era of traffic big data. This traffic flow rate forecasting and prediction of congestion is dependent on the data being collected through these sensors and the relevance of the data being collected e.g., the weather condition, vehicles, vehicle speed, alternative routes, time etc.

The previously used techniques cannot perform well with the changing traffic conditions, therefore we need most recent techniques that can adapt well with the varying state of traffic, deep learning is one data driven approach we can rely on [8], as it has the ability to automatically extract deep traffic features at different levels.

Revised Manuscript Received on January 03, 2019.

Wafa Shafqat, Department of Computer Engineering, Jeju National University, Jeju-do, 63243, Republic of Korea,

Sehrish Malik, Department of Computer Engineering, Jeju National University, Jeju-do, 63243, Republic of Korea,

Yung-cheol Byun, Department of Computer Engineering, Jeju National University, Jeju-do, 63243, Republic of Korea,

Do-Hyeun Kim, Corresponding author, Professor, Computer Engineering Department, Jeju National University, Republic of Korea,

We can find different areas where deep learning is performing exceptionally well e.g., audio/video classification [8]. Due to the dynamic and nonlinear nature of traffic flows, it's not a must to have prior knowledge for the deep learning model to extract and learn the deep features. For traffic flow forecasting, the deep learning method has drawn a lot of research interest [9]. For example, Lv et al. [6] showed a novel deep learning-based traffic flow rate forecasting method, which used weighted auto encoder model to learn traffic flow features. Deep learning is a procedure of machine learning which offers decent instant estimates of the flows of traffic by taking advantage of the reliance.

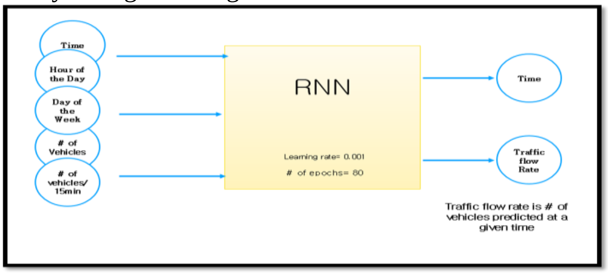

Figure 1. Basic Architecture for traffic flow predictions

Figure 1 shows the abstract illustration of the presented work. It gives an abstract idea of the model as well.

The rest of the paper is organized as follows: section 2 presents the literature review. Section 3 is about the proposed model and architecture. Section 4 is related to the implementation of the proposed idea; describing the data, variables and its preprocessing. Section 5 presents the experimental results and section 6 concludes our work.

II. RELATED WORK

In the transportation network's state of the art, the anticipation and predictions of traffic flows has been targeting time to time and has a good work history. An online learning short term traffic flow predictions model based on support vectors representations, is being introduced by Castro-Neto et al. [6]. There have been some survey studies as well, comparing different existing approaches for short term traffic flow predictions (Lippi et al.) Chan et al. presented an optimized ANN model short-term traffic flow forecasting using a hybrid method [7]. Sun et al. presented a Bayesian network-based method for short-term traffic flow predictions [6]. Lately, deep learning also has been extensively applied to traffic pattern recognition and traffic flow forecasting [9-10]. Additionally, convolutional neural networks help us to extract features in depth [2]. Deep learning techniques which combine different deep learning models in order to take advantage of the strengths of each model, are being used lately e.g., combination of LSTM and CNN, LSTM and RNN etc., all these multimodal techniques have established an growing importance in the primarily in the field of image captioning and image classifications. These multimodal deep learning methods have also been focused by many studies which aim to improve the performance of deep learning methods [3, 2, 5]. These methods are considered to be more powerful and relevant when we have to measure and extract data from multiple sources e.g., traffic flow, density, accidents, weather conditions etc. Such multimodal deep learning methods lack implementation studies for traffic prediction related research. Though, our methodology focuses on the deep learning models which have touched by previous studies for similar sort of experiments, we aim to find an alternative approach for the traditional machine learning and traffic prediction models for traffic congestion analysis. The method proposed, aims to extract the temporal and spatial effects of data, but also make use of the multimodality traffic data (e.g., traffic speed, traffic flow, weather, accidents and traffic journey times, etc.) by multimodal deep fusion learning.

There are some methods that are based on parametric approaches, and some are based on analytical approaches. In the literature survey, we come across many studies taking advantage of neural networks that have been widely applied traffic related predictions in the previous few years. As Artificial Neural Networks (ANN) are capable of controlling multi-dimensional data and due to robust learning capacity, they are being considered a suitable alternative for traffic flow rate predictions [13]. The popular area of machine learning known as Long short term recurrent neural networks (LSTM-RNN) has been focused in many studies. These neural networks are capable of extracting deep and wide meanings inside the data. RNNs are very useful when there are time dependencies in the dataset. We can also find comparative studies in the literature which compares the performance of these neural nets with other deep learning topologies, LSTM RNNs have proved to exhibit better performance in terms of accuracy and firmness.

III. PROPOSED TRAFFIC FLOW PREDICTIONS USING RNN

Our model consists of three basic modules namely input module, the system module, and the output module as shown in figure 2. Input module is to get the input data, apply some preprocessing on it and prepare it to be used for system's training. In system module, parameters are being defined along with the methodology we are applying. RNN are used to train the model and finally in the last module we are forecasting the traffic flow rate at a particular instance of the time.

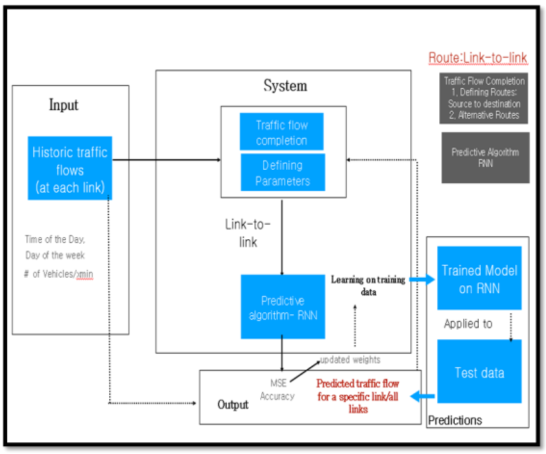

Figure 2. Detailed configuration diagram of traffic flow predictions using RNN

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Hyper parameter optimization is one of the most crucial tasks in modeling a deep neural net. A lot of parameters have to be set for any deep neural network. The most important and key parameters are wright initialization and learning rate. Keras has default values for these parameters. In order to avoid over fitting, in all connected layers, a dropout with probability of 0.2 is being used along with ReLU activation function.

IV. IMPLEMENTATION OF TRAFFIC FLOW PREDICTIONS USING RNN

The following table 1, exhibits the implementation environment of our system. As shown in table 1, we have worked on Mac OS operating system. The programming environment was Python version 3.6. The basic libraries used include Tensor flow, Keras and Scikit-learn. These libraries were useful for the experiments.

Table 1: Implementation Environment

Components	Version
Operating System	Mac OS Sierra 10.12.4
Programming Language	Python 3.6
Libraries	Tensor flow Keras 2.1.3 Scikit-learn 0.19

4.1 Data Description

The first step is to analyze the data and prepare it to be used for training purposes. The most important parameter of the input data is flow rate. The flow rate denoted by 'Q' is defined as the number of vehicles (n) in total, passing through an elected or chosen route in a particular time instance (t). The most widely units are Units usually total number of vehicles per hour. Traffic volume or density typically refers to flow in an hour. As we know that traffic flow is constantly varying. It's very important to accurately select a time interval for which we want to predict the flow. An interval of an hour or more than hour is sometimes too long especially for busy roads. Therefore, we selected a time interval of 15 minutes and 30 minutes in order to check and compare the results and traffic flow accuracy.

$Q = n / t$

The input data contains time stamp, # of vehicles, and flow rate.

Table 2: Input data description

Data set type	Multivariate, Time series
Feature type	Real
Tasks	Forecasting
No. of features	5

Data preprocessing is one of the main challenges while building up a model in machine learning and trying to fit the raw data into this model. For this, an appropriate knowledge and understanding of the problem is required.

Data preprocessing itself is a vast field, there are many steps that can be performed on data based on the data nature and problem domain. First of all, data itself contains a lot of noise e.g., irrelevant fields of data, missing values, null values etc. We remove this noise from data. Then data normalization is the most important preprocessing step where we bring all the data values under one range.

In the learning process, model training is the most important step. In this step the above designed model is trained. We have used 70% of our data as training data and the remaining 30% as of our testing data and validation data. We have used Softmax activation function.

Figure 3 shows a code snippet of model training, we have used mean square error (MSE) as our performance metric. Model defines number of epochs and perform data partition into training, testing and validation sets.

```
def train(models, X_train, y_train, name, config):


    temp = X_train
    # early = EarlyStopping(monitor='val_loss', patience=30, verbose=0, mode='auto')

    for i in range(len(models) - 1):
        if i > 0:
            p = models[i - 1]
            hidden_layer_model = Model(input=p.input,
                                       output=p.get_layer('hidden').output)
            temp = hidden_layer_model.predict(temp)

        m = models[i]
        m.compile(loss="mse", optimizer="rmsprop", metrics=['mape'])

        m.fit(temp, y_train, batch_size=config["batch"],
              epochs=config["epochs"],
              validation_split=0.05)

        models[i] = m

    saes = models[-1]
    for i in range(len(models) - 1):
        weights = models[i].get_layer('hidden').get_weights()
        saes.get_layer('hidden%d' % (i + 1)).set_weights(weights)

    train_model(saes, X_train, y_train, name, config)
```

Figure 3. Model training code snippet

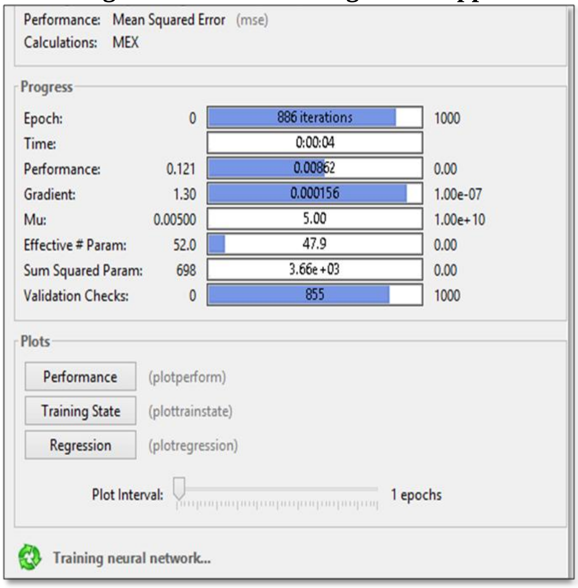

Figure 4. Learning in progress (status bars)

Figure is showing the progress bar of our model during its learning. It is showing the status of the model at current point, e.g., in the figure below we can see out of 1000 epochs, 886 iterations have been performed. At each epoch effective number of parameters is calculated, gradient and mean square error is also calculated.

V. EXPERIMENT RESULTS

Once the model is training, we have used the following figure 5 as a compact representation of the model in order to test it. For the simulations we have trained our network with 10 hidden layers. We have used softmax function as our activation function.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

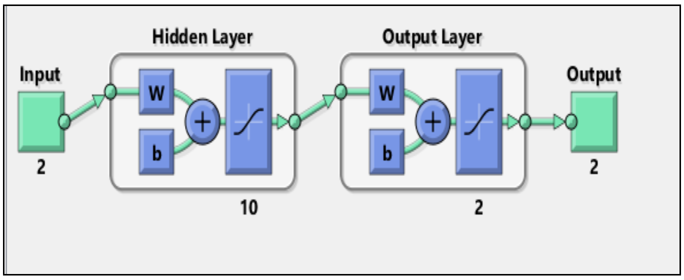

Figure 5. Neural network model

The experimental results, show that our model with LSTM-RNN, have performed better under same environmental conditions, as compared to other neural network models.

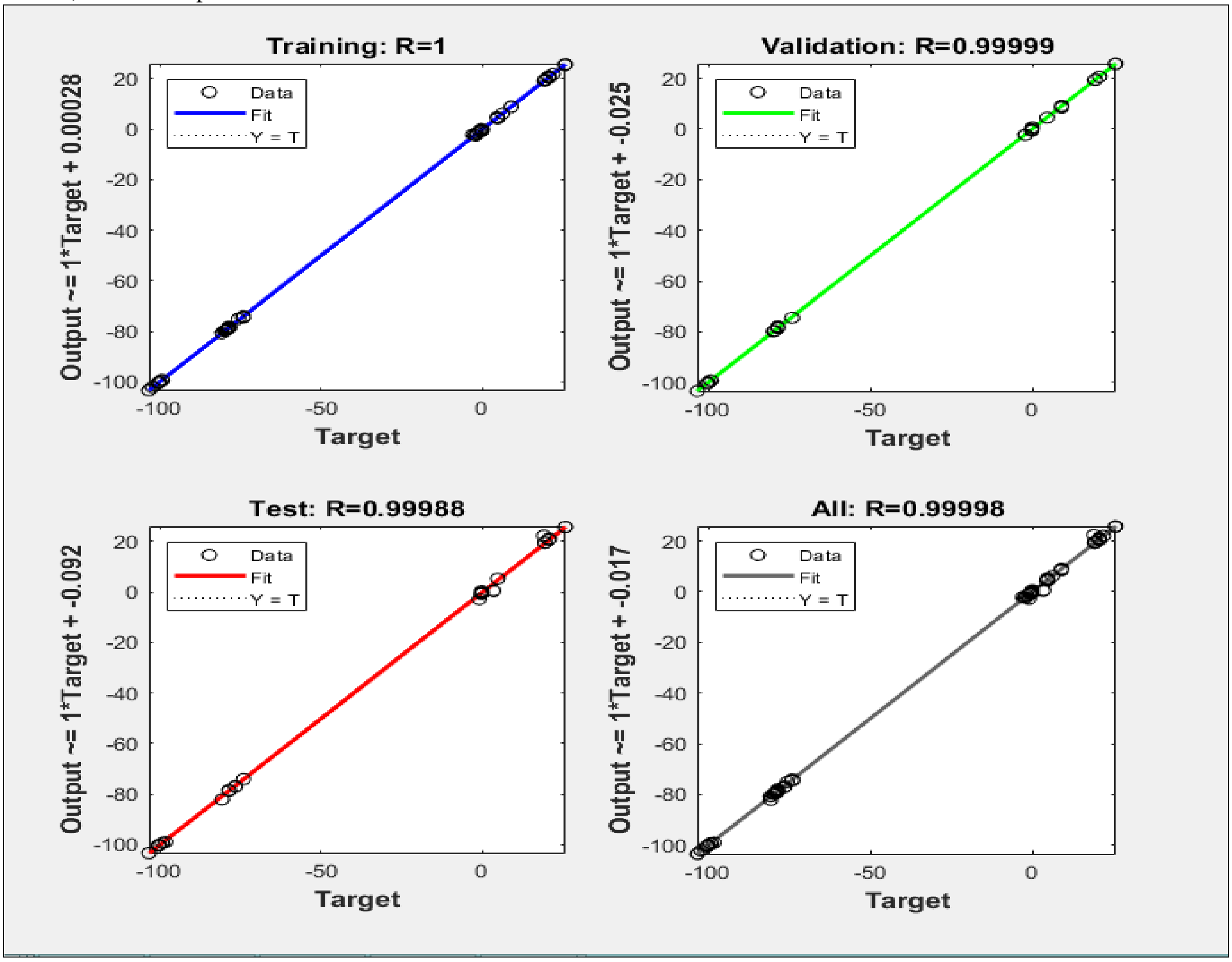

Figure 6. Representation of the training and testing results

Figure 6 is also the representation of the training and testing results where test represents the values that are being predicted and target values are the actual values observed.

Table 3: Prediction performance comparison (Mean Square error)

Model	RMSE
RNN	2.09
Other model	9.01

Table 3 presents the prediction performance of an RNN with 10 hidden layers with a simple neural network model. We can see the training accuracy of RNN is much better than the other network model.

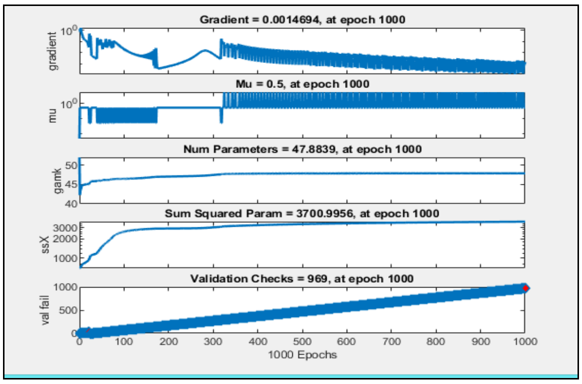

Figure 7. Parameter values in 1000 epochs

Once the iterations are completed, we plotted the results for each model parameter e.g., in figure 7, gradient and number of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

parameters at each epoch are calculated, and then averaged along with the validation checks. It can be observed from the figure that gradient initially was unstable and high, it but towards the end of the epochs, it got stable and reduced as well. Similarly, the number of parameters initially considered was high, more than 50 parameters, but at each epoch, our model has discarded the parameters which are not performing well or have no impact on the performance of the model.

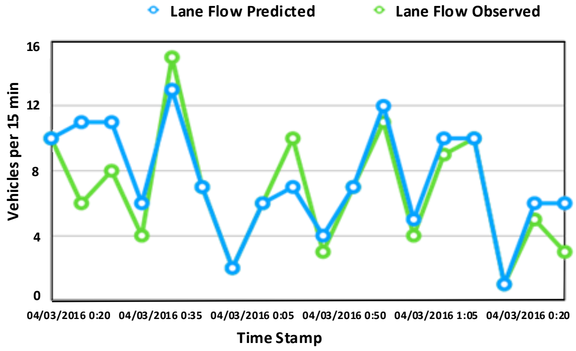

Figure 8. Prediction of Flow rate in every 15 minutes

As we have already seen above the model performance, we have used traffic flow data for the past 2 hours and have predicted flow rate for the next 20 minutes.

As shown in figure 8, we have mapped few of our samples and we can see that our predicted values are getting closer to the actual observations in the data set. Table 4 presents the prediction accuracy of the flow rate of traffic, for 15 minutes and 30 minutes time intervals.

Table 4: Prediction Accuracy of Flow Rate

Time	Percentage Accuracy
15 min	78%
30 min	73%

VI. CONCLUSION

In this era of rapid development and fast moving world, we need to have effective and accurate predictions in every field of our lives to save time and resources. Traffic prediction is also one of the crucial issues and need to be focused on more deeply. In this work, we have used RNN based deep learning technique to predict traffic flow rate at a specific time. From the data set we have computed the traffic flow rates for specific time intervals e.g., predicting traffic flow (i.e. # of vehicles at some time x), in next 15 minutes. For experiments purpose we have used aggregated data for 15 minutes and 30 minutes. We can say that for shorter period of time, predictions can be better comparatively.

Our experiments show that in comparison with the traditional forecasting methods, deep learning methods are expected to deliver substantial enhancements. It can also be inferred that with the number of features expanded, prediction accuracies can be improvised.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(No.2018-0-01456, AutoMaTa: Autonomous Management framework based on artificial intelligent Technology for adaptive and disposable IoT)), and this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2018R1D1A1A09082919). Any correspondence related to this paper should be addressed to DoHyeun Kim; kimdh@jejunu.ac.kr.

REFERENCES

1. The number of cars worldwide is set to double by 2030 [Internet]; [updated 2017 Aug 15; cited 2018 Aug 30]. Available from: https://www.weforum.org/agenda/2016/04/t he-number-of-cars-worldwide-is-set-todouble-by-2040 (website)
2. Kanestrøm PØ. Traffic flow forecasting with deep learning(Master's thesis, NTNU).
3. Jia Y, Wu J, Xu M. Traffic flow prediction with rainfall impact using a deep learning method. Journal of advanced transportation. 2017;2017.
4. Sun S, Zhang C, Yu G. A Bayesian network approach to traffic flow forecasting. IEEE Transactions on intelligent transportation systems. 2006 Mar;7(1):124-32.
5. Huang W, Hong H, Li M, Hu W, Song G, Xie K. Deep architecture for traffic flow prediction. InInternational Conference on Advanced Data Mining and Applications 2013 Dec 14 (pp. 165-176). Springer, Berlin, Heidelberg.
6. Lv, Y., Duan, Y., Kang, W., Li, Z. and Wang, F.Y., 2015. Traffic flow prediction with big data: A deep learning approach. IEEE Trans. Intelligent Transportation Systems, 16(2), pp.865-873.
7. Wu CH, Ho JM, Lee DT. Travel-time prediction with support vector regression. IEEE transactions on intelligent transportation systems. 2004 Dec;5(4):276-81.
8. Schmidhuber J. Deep learning in neural networks: An overview. Neural networks. 2015 Jan 31;61:85-117.
9. Castro-Neto M, Jeong YS, Jeong MK, Han LD. Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert systems with applications. 2009 Apr 1;36(3):6164-73.
10. Lippi M, Bertini M, Frasconi P. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Transactions on Intelligent Transportation Systems. 2013 Jun;14(2):871-82.
11. Abadi A, Rajabioun T, Ioannou PA. Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data. IEEE Trans. Intelligent Transportation Systems. 2015 Apr 1;16(2):653-62.
12. Williams BM, Hoel LA. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. Journal of transportation engineering. 2003 Nov;129(6):664-72.
13. Vlahogianni, Eleni I., Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we're going." Transportation Research Part C: Emerging Technologies 43 (2014): 3-19.