Data analytics for traffic flow prediction in Custom using Long Short Term Memory (LSTM) networks

Data analytics for traffic flow prediction in Custom using Long Short Term Memory (LSTM) networks

Lee Pin Loon[1], Elbaraa Refaie[1], and Ahmad Athif Mohd Faudzi[1,2]

[1]School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, 81310, Johor Bahru, Malaysia.
[2]Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, 54100, Kuala Lumpur, Malaysia.

Corresponding email: athif@utm.my

Abstract. This paper proposes data analysis for traffic flow prediction of customs to help the officer in Customs, Immigration, and Quarantine (CIQ) Complex to understand more about the traffic situation in CIQ. Currently in CIQ, the traffic behaviour for car is unpredictable; sometimes the traffic is very heavy while there are times where all the lanes are cleared. There is a plan to have installation of cameras for smart traffic management system in the future. Therefore, this research aims to have prediction of traffic flow based on time and visualize the trend of traffic data for the officer. The data consist of traffic flow and the respective timestamp. To analyse it with time-series data, Long Short-Term Memory (LSTM) Recurrent Network is used as deep learning approach for prediction. The data pre-processing and training of model would be done using Python. To organize the data, Tableau Prep Builder is used and integrate with Python to publish the data to Tableau Server for storage. An interactive dashboard would be designed on Tableau and made available online for the usage of the officer.

1. Introduction

From estimation of officials, there are approximately 145,000 vehicles that pass-through Customs, Immigration and Quarantine (CIQ) complex in Johor Bahru every day. BSI CIQ (Bangunan Sultan Iskandar CIQ) has become one of the most hectic border in the world when 300,000 people crossing the border daily. Normally for weekdays, 1 to 2 hours would be the duration it takes to commute from Johor Bahru across the Causeway bridge of 1 km-long [1]. If it is on weekends or public holiday, the duration would be even longer. The former Prime Minister, Tun Dr. Mahathir Mohamad has mentioned that it has become a priority for Malaysia to come out with a solution for the traffic congestion at the Malaysia-Singapore borders.

To resolve the congestion, traffic flow prediction is a very vital component. The forecasting of characteristics of road traffic such as occupancy, flow and speed would contribute to the planning of new road networks, modifications to existing strategies [2]. Precise traffic flow forecasting would greatly help in reducing traffic congestion, improving the air quality, and provide references to government in making decisions.

As traffic flow is stochastic, an accurate traffic prediction is a challenging issue. For these past years, it has been a trend to use statistical methods and Artificial Intelligence (AI) techniques in order to analyse and predict the road traffic characteristics. There are three categories of current existing prediction

schemes that are namely parametric methods, nonparametric methods, and hybrid methods. Parametric methods include Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average method (SARIMA) [3], and Kalman filter [4]. Some of the instances of nonparametric methods include K-Nearest Neighbours (KNN) and Support Vector Regression (SVR) while hybrid methods are combination of parametric and nonparametric methods. The algorithm that would be discussed below would be ARIMA, KNN, and SVR.

2. System Design

The approach for the data analysis of this project consisted of data classification, data preprocessing, parameter settings and evaluation of traffic prediction model. Some important and useful libraries are included in Python code. Tableau software would be discussed also as a powerful data visualization tool in Business Intelligence Industry. Data analysis can be fast by sampling complex data into understandable format. **Figure 1** shows the prediction process using LSTM prediction.

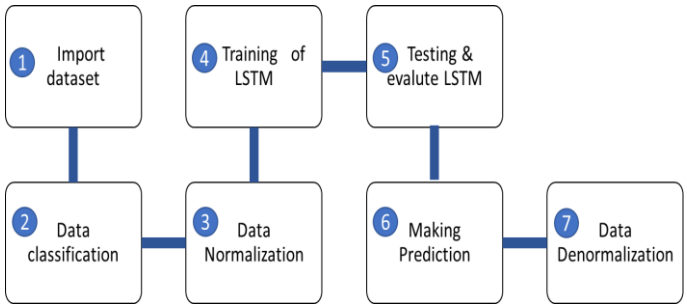

Figure 1. Overall steps in LSTM prediction.

2.1 Data classification

To increase the accuracy and relevance of data, the traffic flow data would be classified into two categories, which are weekdays and weekend. The weekend categories for the traffic flow in custom would cover Saturday and Sunday while the weekday's category would include from Monday to Friday. There are two models to be trained in total; each category would have a trained model.

2.2 Data pre-processing

Before the data undergoes training or inserted into LSTM neural network, they have to be preprocessed. There are few elements in data preprocessing which are traffic flow calculation and normalization of data. The machine learning libraries used in Python are Scikit-learn and Keras together with Tensorflow as backend. NumPy and Pandas libraries are also used as well. NumPy is used mainly for mathematical computation on array and matrices while Pandas provides DataFrame that is like a spreadsheet containing column names and row labels.

2.2.1 Traffic flow unit

For this project, the traffic flow would be the number of vehicles passed through in 5 minutes. Hence, the traffic flow would be generated in the unit of vehicle/ 5 minute. The interval of 5 minute would be constructed and the traffic flow would be preprocessed. The reason why the interval chosen is 5 minutes is due to the references from others [5], [6]. They have aggregated the data into 5 minutes of interval, and they have mentioned that 5-minute traffic flow is more suitable and predictable.

2.2.2 Normalization of data

Input for the neural network normally has to be normalized or scaled before they undergo training. This is because the unscaled data which may have wide range would reduce the speed of learning. It may also decrease the convergence of the model which would make the network unable to effectively learn. There

are basically two types of scaling of the series of data which are normalisation and standardization. For this project, normalisation method is used to scale the input.

2.2.3 Parameter setting
The hardware used for this project is Intel i5-7200U Central Processing Unit (CPU) and a NVIDIA GeForce 940MX graphics Graphics Processing Unit (GPU). The activation function used is ReLU function, the number of hidden units is [64, 64], the batch size is 64 and the timestamp is 24 and the output layer dimension is 3.

2.3 Model design
Data set is collected from the vision system. The system used Yolov4 for vehicles classification and detection. Data is collected and stored in the Excel sheet with timestamp to be fed to our model. Next, data will be divided to training set and validation/testing set by 80% and 20% respectively. The training dataset is fed to train the model while the validation dataset is used for model verification and testing model's accuracy. Dataset is fed with timestamp and max count for each set which helps in data normalization. For optimal performance data is normalized between 0, 1 using the max count using Scikit- Learn's MinMaxScaler .

LSTMs expect our data to be in a specific format, usually a 3D array. We start by creating data in 24 timesteps and converting it into an array using NumPy. Next, we convert the data into a 3D dimension array with Train samples, 24 timestamps, and one feature at each step.

To construct out model, we mainly based on Keras to add the layers to our model. To prevent overfitting issue from happening, Dropout layer is added to the model with 0.2 (20% of the input layers will be ignored) for better output results. LSTM layer added with parameters in 2.2.3. After drop out a dens layer to set a 1-unit output. Using Adam optimizer and MSE (Mean Squared Error) as a loss function, we could compile the model. After model is being compiled, it is fitted to run 60 epochs (normally 50 – 100 is fine) with 64 batch sizes. When the training process is over, we could save our model to reuse it in the prediction process.

2.4 Evaluation of traffic prediction model
There are few parameters used to evaluate the traffic prediction model that are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square and Error (RMSE).

2.4.1 Mean Absolute Error (MAE)
MAE illustrates the magnitude of the residual. Each residual is proportionally to the total error. MAE acts as a protection against outliers.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad (1)$$

2.4.2 Mean Absolute Percentage Error (MAPE)
MAPE is ratio of the sum of the individual absolute errors to by the demand (each period separately) as average of the percentage errors.

$$\text{MPE} = \frac{100\%}{n}\sum\left|\frac{y-\hat{y}}{y}\right| \quad (2)$$

2.4.3 Root Mean Square Error (RMSE)
The goal of RMSE is to generate prediction that is correct for the average.

$$\text{RMSE} = = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \quad (3)$$

2.5 Tableau Software

Tableau Prep Builder is used to integrate the Python scripts for data pre-processing and prediction. Tableau is able to connect to datasources published on cloud storage in Tableau Server and update the datasources. There are two main processes in Tableau Prep Builder: training and prediction process. Python script is embedded in those block diagrams which look like note. **Figure 2** below shows the training process. First, the TrafficData datasource containing all the traffic flow data would undergo step to add in column of "Day" based on the date. Then, the data would be arranged and spilt into weekday's data and weekend's data. Each path experiences the same process for training, creating of model and uploading of model. Lastly, both the training data of weekdays and weekends would be uploaded to Tableau Server. The training data stored in Tableau Server is for the usage of getting the scaler for normalisation and denormalisation of data when the prediction processed is executed.

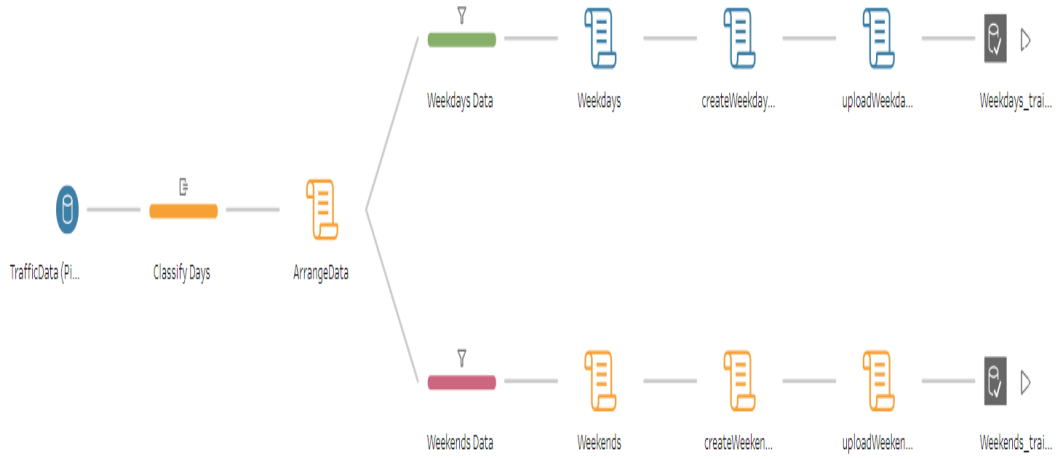

Figure 2. Training process in Tableau Prep Builder.

3. Results and discussion

The raw data from online source are in comma-separated values (csv) format. Thus, Pandas library is used to read the csv file using build in function named read_csv. After the file is read, DataFrame is created to help in organizing the data with column names as shown in **Figure 3**.

	5 Minutes	Flow (Veh/5 Minutes)
0	04/03/2019 0:00	16
1	04/03/2019 0:05	10
2	04/03/2019 0:10	11
3	04/03/2019 0:15	11
4	04/03/2019 0:20	6
5	04/03/2019 0:25	13
6	04/03/2019 0:30	7
7	04/03/2019 0:35	2
8	04/03/2019 0:40	6
9	04/03/2019 0:45	7
10	04/03/2019 0:50	4

Figure 3. Creation of Data Frame.

From the DataFrame shown, we can see that there are 2 columns of data. The first column is "5 Minutes" which represents the time for each 5 minutes. The timestamp is sliced into 5 minutes which means the data represented would have a sampling rate of 5 minutes. In other words, for each single point

of data, the interval range between them is 5 minutes. It would be shown and reflected using graphical method in the next section of this chapter. The second column of data is "Flow (Veh /5 Minutes)" which represents the traffic flow in which the unit is vehicles per 5 minutes. In other words, the column represents the number of vehicles passed through within the interval of 5 minutes according to the time illustrated in column "5 Minute" as shown in **Figure 4**.

After the creation of DataFrame, the traffic flow data has to be normalised. The normalisation is done using scikit-learn , MinMaxScaler. The scaling for normalisation would convert the maximum value from dataset to 1 and minimum value to 0 while the rest of the data would be weighted accordingly and result in value between 0 and 1. From Fig. 3, the maximum value of traffic flow among the samples of data is 16 vehicles per 5 minutes while the minimum value of traffic flow data is 2 vehicles per 5 minutes. Thus, from Fig. 4 below, we can observe the normalised DataFrame, the value of 16 vehicles per 5 minutes has been normalised to 1 while the value of 2 vehicles per 5 minutes has been normalised to 0.

	5 Minutes	Flow (Veh/5 Minutes)
0	04/03/2019 0:00	1.000000
1	04/03/2019 0:05	0.571429
2	04/03/2019 0:10	0.642857
3	04/03/2019 0:15	0.642857
4	04/03/2019 0:20	0.285714
5	04/03/2019 0:25	0.785714
6	04/03/2019 0:30	0.357143
7	04/03/2019 0:35	0.000000
8	04/03/2019 0:40	0.285714
9	04/03/2019 0:45	0.357143
10	04/03/2019 0:50	0.142857

Figure 4. Normalized Data.

To improve the length of predicted duration, some trials have been done by tuning the size of input and output data parameters for LSTM model to reach the best prediction results. The logs of input and output dimension has been tested to produce the best result. The figures (**Figure 5, Figure. 6, and Figure. 7**) and tables (**Table 1, Table 2, Table 3**) below show the results of different number of input and output dimension for LSTM model.

Table 1. Evaluation of predication model using one hour input to produce one output

Evaluation parameters	Value
MAE	33.744938
MAPE	21.252576%
RMSE	49.826746

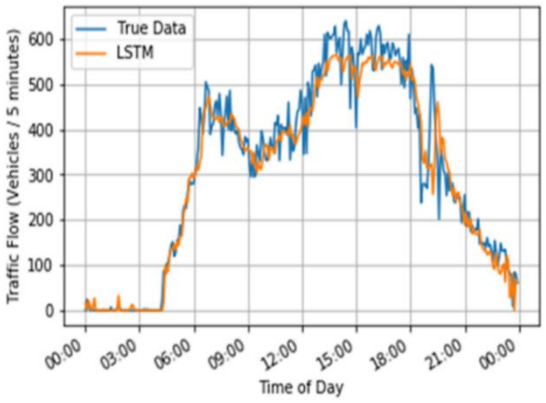

Figure 5. Validation of prediction model using one hour input to produce one output.

Table 2: Evaluation of prediction model using two-hour inputs to produce three outputs

Evaluation parameters	Value
MAE	33.700711
MAPE	19.948792%
RMSE	49.612709

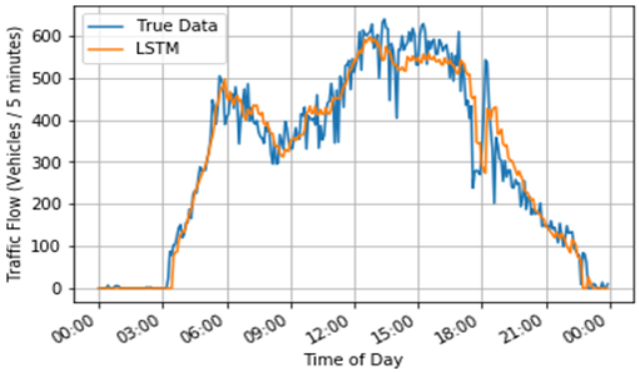

Figure 6. Validation of prediction model using two hour inputs to produce three outputs.

Table 3: Evaluation of prediction model using two-hour inputs to produce four outputs

Evaluation parameters	Value
MAE	36.003413
MAPE	17.092985%
RMSE	55.837406

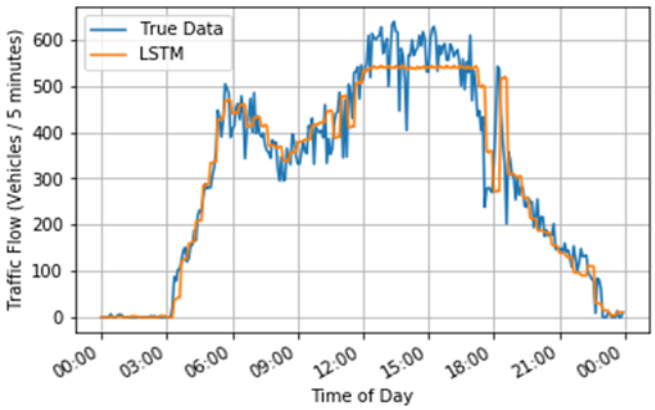

Figure 7. Validation of prediction model using two hour inputs to produce four outputs.

From the figures above, we can observe the best parameters would be having two hours input to produce three outputs as the result of overall evaluation is the best. In other words, the final model used is having logs of 24 as each log of 12 represents one hour, while the predicted output is up to the next 15 minutes as each output step represents value for the next 5 minutes. Further increase in output dimension higher than three causes the accuracy of model to drop. As shown in **Figure 6**, the predicted traffic flow could not follow the changes in actual traffic flow due to the stochastic and dynamic characteristic of traffic flow. There are many possibilities that could have happen within the timeframe of 20 minutes of prediction ahead. Thus, the final prediction model used is using two hours inputs to predict the traffic flow of next 15 minutes.

The prediction process in Tableau Prep Builder can be performed over several steps. First, the TrafficData datasources would undergo step to classify the date of traffic flow into days, the same step as in training process. Then, after the data is arranged according to timestamp, it will undergo a script named getDataForPrediction that will crop the latest two hours of traffic flow data to act as prediction input. The Union1 would then union the traffic flow data together with the training data of weekends and weekdays before they are input into prediction script named Predict. The script will apply the LSTM model for prediction of the traffic flow of the next 15 minutes. After the prediction script, it will be added for column days and joined with a datasource named PredictionTable. The joining process would join the old PredictionTable with latest predicted value of traffic flow before they are updated, arranged and published to PredictionTable datasource. The Prediction Table datasource in other words would always have traffic flow data of 15 minutes ahead of the TrafficData data source.

Figure 8 illustrates the current tab of dashboard. The current tab of dashboard would show the traffic flow data of the latest day in TrafficData datasource. There are two lines on the graph coloured in orange and blue. Orange line represents the actual traffic flow while blue line represents the predicted traffic flow. Hence, the predicted traffic flow is always 15 minutes ahead of the actual traffic flow. For the lower section of the current tab of dashboard, there is a section labelled as peak hour where user can choose the number of peak hour they wish to view. The value for traffic flow and time for peak hour would be displayed.

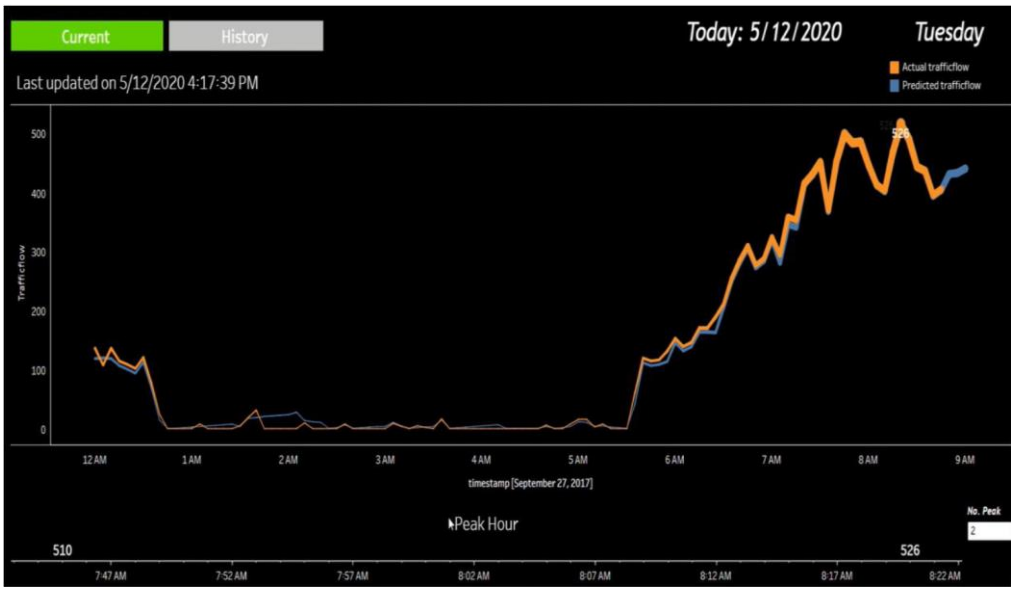

Figure 8. Current tab of dashboard.

For the history tab of dashboard as shown in **Figure 9**, it allows the user to view the trend of historical traffic flow. There is a date picker at the right side of history tab of dashboard for user to pick a date to view the historical trend of traffic flow. After a date is picked, few radio buttons can be selected to view the dates related to the selected date such as yesterday, last week and last 2 weeks. The peak hour viewer at lower section is having the function same as in current tab, showing the time and value of peak traffic flow of the selected date.

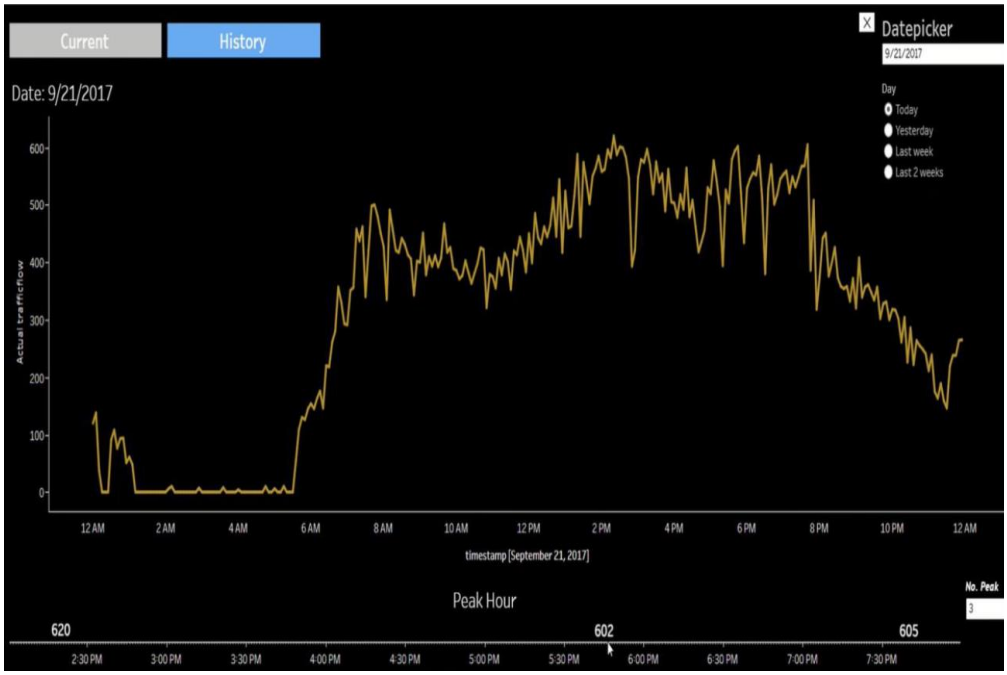

Figure 9. History tab of dashboard

4. Conclusion

The expected outcome of this data analytics for traffic flow prediction of customs is to assist the officers in CIQ. Using 3 hours input data to produce output predictions for the next 15 minutes can generate output with accuracy up to 83%. This project is using deep learning approach that is LSTM to predict the traffic flow based on historical data. This can help in visualize the trend of traffic flow. The results are satisfied and this project would be part of the development for traffic management project in CIQ.

Acknowledgements

The research has been carried out under program Research Excellence Consortium (JPT (BPKI) 1000/016/018/25 (57)) provided by Ministry of Higher Education Malaysia (MOHE). The author also would like to thank Universiti Teknologi Malaysia (UTM) for work and facilities support.

References

[1] Clearing The Causeway. (n.d.) Channel News Asia. Retrieved December 30, 2019, from https://infographics.channelnewasia.com/interactive/causewayjam/index.html

[2] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri 2019 *Smarter traffic prediction using big data, in-memory computing, deep learning and gpus*, vol. 19, no. 9.

[3] D. Miao, X. Qin, and W. Wang 2014 The periodic data traffic modeling based on multiplicative seasonal ARIMA model, *2014 6th Int. Conf. Wirel. Commun. Signal Process. WCSP 2014*, pp. 1–5, doi: 10.1109/WCSP.2014.6992053.

[4] S. V. Kumar 2017 Traffic Flow Prediction using Kalman Filtering Technique," *Procedia Eng.*, vol. 187, pp. 582–587, doi: 10.1016/j.proeng.2017.04.417.

[5] R. Fu, Z. Zhang, and L. Li 2016 Using LSTM and GRU neural network methods for traffic flow prediction, *Proc. - 2016 31st Youth Acad. Annu. Conf. Chinese Assoc. Autom. YAC 2016*, pp. 324–328, doi: 10.1109/YAC.2016.7804912.

[6] W. Wei, H. Wu, and H. Ma 2019 An autoencoder and LSTM-based traffic flow prediction method," *Sensors (Switzerland)*, vol. 19, no. 13, pp. 1–16, doi: 10.3390/s19132946.