Traffic Volume Prediction: A Deep Learning Approach

Hitesh Thadhani, Prankur Garg
School of Information Studies, Syracuse University
Syracuse, NY, USA
{hthadhan, pgarg03}@syr.edu

ABSTRACT

Short term traffic flow prediction is essential for building a robust Intelligent Transportation System(ITS). Traffic volume prediction helps commuters make an informed choice about the routes, reducing their travel time, and helping the government to allocate more resources on busy routes to decongest the traffic. Recently, deep learning models have shown promising results in predicting traffic volume by capturing the complex nonlinear dependencies. In this paper, we conducted multiple experiments on the Metro Interstate Traffic Volume dataset of Hourly Interstate 94 Westbound traffic volume for MN DoT ATR station 301[1] using Feedforward Neural Networks, Recurrent Neural Networks (RNN) and their variations Long Short-Term Memory(LSTM) and Gated Recurrent Units(GRU) for both Univariate and Multivariate Time series forecasting which have memory to learn from long past sequences to predict future volume accurately. We conclude by comparing these deep learning models along with Convolutional Neural Networks(CNN) experiments and which one works the best in predicting traffic volume along with the challenges faced and future scope of the project.

KEYWORDS

Traffic Volume Prediction, Feedforward Neural Networks, Recurrent Neural Networks, Long Short-Term Memory, Gated Recurrent Unit, Convolutional Neural Networks, Time Series

1. INTRODUCTION

Short-term traffic states prediction aims at forecasting traffic conditions, e.g., travel times, traffic speeds, and traffic volumes on road segments for a certain future time period, typically less than an hour.[2] In the real world, accurately predicting the traffic flow volume has many applications. For example, taxi demand prediction can help taxi companies pre-allocate taxis; traffic volume prediction can help transportation department better manage and control the traffic to ease traffic congestion.[3]

Traditional traffic volume prediction methods for time series forecasting using autoregressive(AR), autoregressive integrated moving average (ARIMA) have been used for decades. These traditional methods work on capturing linear relationships between the time series data and are mostly used for univariate time series forecasting. Deep learning methods such as RNN’s and their variants are able to model the complex nonlinear dependencies along with accounting for other external factors such as local events or national holidays, weather conditions which influence the accurate traffic volume prediction.[3]

We used Feedforward Neural Networks to check which features help in a model's prediction including weather, holiday, weekend along with finding if feedforward neural networks can capture the time series information. Univariate time series forecasting is done by building deep learning models. We train RNN's and their variants on these identified features for Multivariate Time series forecasting. We have used Mean Squared Error (MSE), Computational Effort required for prediction along with visualizing the actual prediction to compare different algorithms for their effectiveness in traffic volume prediction.

Section 2 discusses the relevant previous work done in the prediction of traffic volume, problem statement, and data description. Section 3 discusses the deep learning LSTM and GRU. Section 4 discusses the methodology used along with different algorithms used. Section 5 discusses the experiments performed and the results obtained. Section 6 concludes with the conclusion and future work.

2. PRELIMINARIES

2.1 RELEVANT WORK

Traffic Volume Predictions have been carried out for decades to solve the commute time problem. Because of their strength on handling non-linearity and universal approximability of unknown functions, neural network approaches have been frequently employed for traffic flow prediction from earliest researches to today.[4] For example, Zheng et al. (2006) combined neural networks and bayesian inference to forecast future traffic flow[4]. Traffic flow is affected by weather conditions such as thunderstorms, snow, and how these adverse conditions could impact the traffic volume is studied. The research paper **"Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach"** talks about the correlation between the weather and traffic, improving the effectiveness of the prediction of traffic volume by accounting for weather data.[5]. Another research that talks about the impact of weather data to predict traffic flow modeled through Gated Recurrent Units (GRU) increases the predictive accuracy and reduces the prediction error.[6]

2.2 PROBLEM STATEMENT

With more than 1 billion cars on roads today, which is expected to double to around 2.5 billion by 2050, designing superefficient navigation and safer travel is becoming a major challenge for transportation authorities.[5] This huge increase in the number of vehicles has increased pollution levels, travel time for the commuters. Accurate traffic volume prediction would provide guidance to the individuals to choose the best time to travel to avoid unnecessary delays due to roadblocks or take alternate routes.

Given the multivariate historic time series data for traffic volume, our project aims to predict the future traffic volume as close as possible to the actual traffic, studying the effect of holidays and weather on the prediction and at the same time reducing RMSE and Computational Effort required to do the prediction. We consider traffic volume until the time t to predict the future volume at time t+1 given the data is hourly.

2.3 DATA DESCRIPTION

The dataset used is Metro Interstate Traffic Volume from the UC Irvine Machine Learning Repository. This dataset is multivariate hourly time series data with other features such as holiday, weather conditions. The data is collected for 6 years from 2012 to 2018 for Minneapolis-St Paul, MN traffic volume for westbound I-94.[1] The dataset has 3 categorical features for holidays, weather main, and weather description which is a detailed version of the weather main. The dataset has numerical weather features for rain, snow, temperature, clouds corresponding to every hourly traffic volume.

3. DEEP LEARNING LSTM and GRU

The basic idea of a recurrent neural network (RNN) is that an output is produced at each time step and the hidden units are recurrently connected, which means the output h_t depends on both the input x_t and previous information h_{t-1}.[7] RNN is suitable for sequential data modeling or time series forecasting for applications such as speech recognition, stock price prediction, traffic volume predictions. Long Short-Term Memory (LSTM) addresses the issue of vanishing gradient, faced by RNN's for the long sequences of data. LSTM consists of a mechanism called gates to allow what and how much past information is carried forward and how much is forgotten. LSTM has 3 gates input gate, output gate, and forget gate with cell state.[8]

GRU's on the other hand consists of only 2 gates update and reset gate with cell state from LSTM being removed. GRU uses less memory and is faster to execute as compared to LSTM with 3 gates.[9]

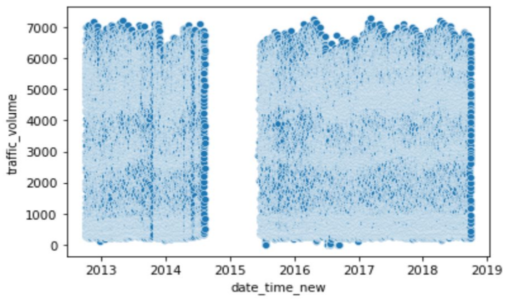
4. METHODOLOGY

4.1 DATA PRE-PROCESSING

Before diving into models, there was a need to pre-process the data.

Missing values

When we ran an exploratory analysis on the data, we found that there were missing values for approx. 8 months' worth of data. This was a challenge because we thought that if the values are removed unchecked, it could affect the sequence of the data which could, in turn, affect the outcomes of the model. Thus, we decided to remove the whole 8 months from the



dataset.

Figure 1: Traffic Volume from 2012 to 2018

Removing Outliers

Then, we found some outliers in the columns temperature and rain. So, we suppressed those values with the previously correct value. We also converted the temperature column values which were in Kelvin to Celcius to better understand the data for ourselves.

Anomaly in holiday column

While looking at the data, we noticed that there was a huge anomaly in the holiday column of the dataset, the problem was that when there was a holiday on a particular day, it would mark that day as a holiday only for the first hour. The remaining 23 hours were counted as non-holiday. The holiday would appear (name of the holiday) for the whole day after anomaly correction.

Label encoding

The holiday column had values that were categorical and had 12 unique holiday values. So, we decided to use one-hot encoding for the holiday column since there was no seasonality with respect to traffic_volume. We converted the categorical values to integer values 0 and 1 (0 - non-holidays and 1 - holidays).

Then, in weather desc. column, we ran a script that would find a particular string in the list of strings name that row according to that particular string. Ex. The row which had snow in it was converted to snow. In this way, we were able to reduce the number of unique values from 35 to 9 in that column.

In one case, for the experimentation, we also did entity embedding for categorical variables holiday and weather description to compare it with other encoding techniques such as label encoding and one-hot encoding.

Column removal

We decided to remove the date column because we had converted it to an index column. We also decided to remove weather main since weather desc. was similar to it having additional information.

Imputing missing values

In some cases, we found that there were missing values for the hours. We tried many methods including linear, polynomial, nearest, time using the backfill. Among these, Time-method worked best for the interpolation of temperature, rain_1h, snow_1h, clouds_all and traffic_volume. For the categorical columns related to weather previous value was used to impute the next missing value as there was no interpolation method available for the categorical weather values.

4.2 APPROACH

4.2.1 FEATURE SELECTION

Features are selected along with the data for a particular time duration based on the experiment being carried out. For an experiment, we may not use holiday or weather data and can use data for a particular time frame.[11]

4.2.2 SUPERVISED TO TIME SERIES DATASET CONVERSION

For feed forward neural networks, we do not need to convert the data to a time series format with lags. The data is fed sequentially to the model.[11]

For RNN deep learning models, the original data is converted to a time series shifted format to be fed to RNN models. Predicting the traffic volume at time t, we would consider the traffic volume at t-1, t-2 etc. based on the lags we define. Historic data is used to predict the current data using the lags.[11]

4.2.3 BUILDING DEEP LEARNING MODELS

After the data for an experiment is finalized, we scale all the features using

MinMax Scaler on a range from 0 to 1 to ensure algorithms are not biased towards the features with higher coefficients and treat all the features equally. The dataset is divided into train, validation and test with a split ratio of 70:20:10. 10% of the data is used for testing which the model has never seen to evaluate the performance of the model. The size of the dataset is not large enough to build a very complex neural network. Models are built initially with a smaller network and then complexity is increased which are then checked if the increase in complexity produces substantial improvement in the results and if the model complexity can be justified. [11]

4.2.4 MODEL EVALUATION

After the experiment is conducted, models are evaluated using RMSE as the metric of measuring model performance. MSE and Loss on the validation data is checked to see how the model is performing. Actual vs Prediction traffic volume is plotted to see how well the model is doing in predicting the unknown test data. Same model is run 5 times and mean RMSE and Standard Deviation of the RMSE is calculated. Computational effort required by the model is calculated to do comparison among various algorithms along with mean and standard deviation of the RMSE on the test data.[11]

5. EXPERIMENTS AND RESULTS

There were a series of experiments conducted by us on the pre-processed dataset for both univariate and multivariate data.

Optimizers

We tried various optimization techniques for our models to see which one gave the best results. These techniques were RMSProp, ADAM and ADAMAX. Out of these, we found that ADAM gave the most favourable results.

Hidden layers and number of nodes

We tried various combinations such as using 1, 2 and 5 hidden layers. The best results we got were with 2 hidden layers at the end after comparison. As for the number of nodes, we kept 10-12 nodes as the results were similar.

Epochs

We ran models with 50 and 100 epochs along with Early Stopping to avoid overfitting when the validation loss is not improving, so we found that 50 epochs are good enough to train our model which would also save computational effort required to run the models for 100 epochs without any substantial improvement in the validation mse and loss.

Dropout

We ran the models with and without dropout and found out that dropout of 0.2 worked the best which reduced the RMSE and helped in giving better prediction results. Dropout is a regularization technique to prevent overfitting by randomly setting the output of the nodes to 0.

5.1 EXPERIMENT I FEEDFORWARD NEURAL NETWORKS

Feedforward neural networks were run with a smaller network given the data size is not large. Multiple experiments were carried out by removing categorical variables, using entity embedding for weather data, binary encoding holiday and weather data, separately running models for holiday and non-holiday data. Mean MSE, RMSE and standard deviation of 5 runs of each experiment are recorded for comparison to determine which is the best model. Input data was scaled and all the metric calculation was done on the scaled version which gives the RMSE, MSE values in the 0.25 range without inverse scaling to the original scale only for feedforward neural network experiments.

	Mean MSE	Mean RMSE	Avg. Computational Effort (epoch*number of weights)
No Categorical features	0.0676±1.21E−6	0.260±0.0011	40*409=16,360
Holiday and Weather data label encoded	0.067±4.9E−7	0.259±0.0007	38*433=16,454
Binary Encoded Categorical Variables	0.0686±4.0E−6	0.262±0.0020	38*433=16,454
Entity Embedding for weather feature	0.0655±1.0E−6	0.256±0.0010	34*613=20,842
1 year data for training and rest for test	0.0681±1.6E−7	0.261±0.0004	25*433=10,825
Predicting Holiday data using Non-holiday data trained model	0.0676±1.44E−6	0.260±0.0012	35*433=15,155

Table 1: Feedforward neural networks experiments comparison for Avg. RMSE, Avg. MSE and Avg. Computational Effort

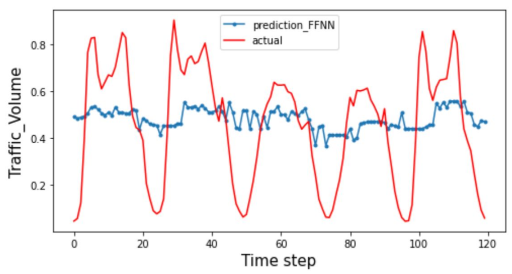


Figure 2: Traffic Volume Actual vs Predicted by Feed Forward Neural Networks

5.2 EXPERIMENT II UNIVARIATE TIME SERIES

For Univariate time series, we performed 2 major variations with all 3 deep learning models(RNN, LSTM and GRU) for a delay of 1 period and delay of 12 periods accounting for 12 hours of lag data to predict the traffic volume at time t. Mean MSE, RMSE and computational effort were used to compare different models for finding the best model.

Among the 2 variations for lags, the model where we predicted traffic volume at time t using data from time t-1 to t-12 for 12 periods gave best results.

The RMSE and MSE for the 3 models with 12 lag periods is shown below.

Metric	rnn_lag12	lstm_lag12	gru_lag12
Avg. Root Mean Square Error	342.56±19.00	347.11±8.28	337.58±1.30
Avg. Mean Square Error	117347.3536	120485.3521	113960.2564
Avg. Computational Effort	50 * 341 = 17,050	45 * 491 = 22,095	45 * 371 = 16,695

Table 2: Avg. RMSE and standard deviation, Avg. MSE and Avg. Computational Effort comparison for RNN, LSTM and GRU with 12 period lags for Univariate Time Series

The graph below illustrates how well the 3 algorithms are performing for predicting the future traffic volume.

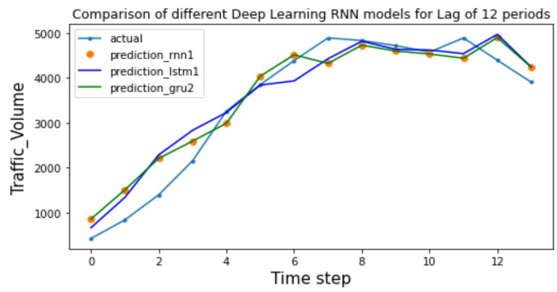


Figure 3: Traffic Volume actual vs predicted by RNN, LSTM and GRU for Univariate Time Series Forecast

For Univariate time series, Convolutional neural networks were also tried to see how CNN's can be used for time series data modeling. CNN with lag of 12 periods was tested as it came out to be the best for all the Recurrent neural network models.

Comparison of CNN vs GRU on the basis of RMSE, MSE and computational effort is shown below.

Metric	gru_lag12	cnn_lag12
Avg. Root Mean Square Error	337.58±1.30	401.23±21.22
Avg. Mean Square Error	113960.2564	160985.5129
Avg. Computational Effort	45 * 371 = 16,695	* 8769 = 438,450

Table 3: Avg. RMSE and standard deviation, Avg. MSE and Avg. Computational Effort comparison for GRU and CNN with 12 period lag for Univariate Time Series Forecast

The graph of future traffic volume prediction for GRU vs CNN is shown below.

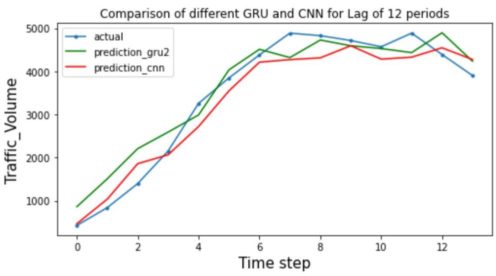


Figure 4: Traffic Volume actual vs predicted by GRU and CNN for Univariate Time Series Forecast

5.3 EXPERIMENT III MULTIVARIATE TIME SERIES

For multivariate, we conducted various experiments with different models (LSTM, GRU, RNN) using various combinations of hyperparameters on the whole dataset.

We ran 4 different ones for each of them. One was predicted using 1 year of data for training, one with basic parameters, one with varying hidden layers and lastly with data shifted by a period of 12 hours. Then, we used RMSE to compare the models. Lastly, we used a predicting plot to compare and get the best model that is suitable for forecasting.

We found that for every model, the model with lag = 12 gave the best results.

	lstm_lag	gru_lag	rnn_lag
Root Mean Square Error	476.987307	472.818923	584.934238
Mean Square Error	227516.890625	223557.734375	342148.062500

Table 4: Avg. RMSE and Avg. MSE for RNN, LSTM and GRU with 12 period lag for Multivariate Time Series Forecast

Then for the same model, we ran it 5 times to get the avg. and standard deviation.

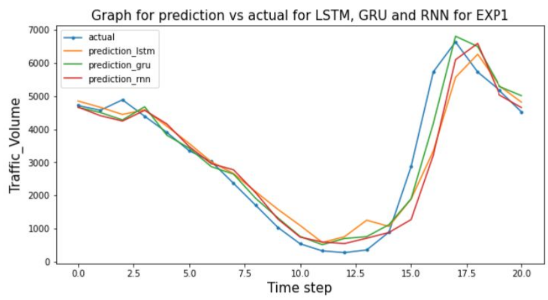


Figure 5: Traffic Volume actual vs predicted by RNN, LSTM and GRU for Multivariate Time Series Forecast

We then repeated the whole process from above and ran a model with dropped columns (holiday and weather desc.) to check if there is a difference in the final result.

GRU LAG			GRU LAG2		
	RMSE	MSE		RMSE	MSE
EXP1	472.819	223557.734	EXP1	535.925	287215.375
EXP2	530.683	299881.594	EXP2	550.088	302596.656
EXP3	475.191	225806.328	EXP3	539.328	290874.938
EXP4	500.684	317770.375	EXP4	554.086	307011.344
EXP5	450.373	202835.891	EXP5	564.926	319141.313
MEAN	*485.950*	*253970.384*	*MEAN*	*548.871*	*301367.925*
ST DEV	*30.708*	*51261.969*	*ST DEV*	*11.674*	*12841.814*

Table 5.1, 5.2: Avg. RMSE and standard deviation, Avg. MSE and standard deviation with individual RMSE and MSE for 5 runs of GRU for 2 experiments EXP1 and EXP2 with and without holiday and weather data

GRU Model for lag = 12	Parameters	Avg. epochs	Computation effort
GRU EXP1	630	40	25,200
GRU EXP2	540	37	19,980

Table 6: Computational Effort comparison for 2 GRU experiments EXP1 and EXP2 with 12 period lag for Multivariate Time Series forecast

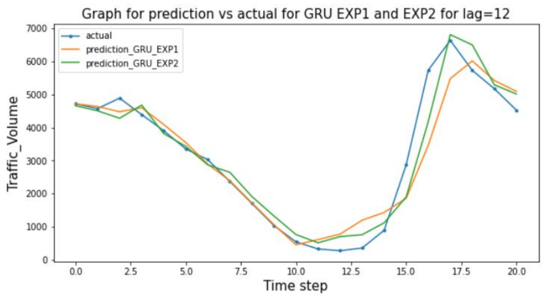


Figure 6: Traffic Volume actual vs predicted by GRU with holiday and weather description data EXP1 and GRU without holiday and weather description data EXP2

6. DISCUSSION

6.1 CONCLUSION

6.1.1 FEEDFORWARD NEURAL NETWORKS

For all the experiments carried out for feedforward networks, the best results came out for Entity Embedding of weather data. Although computationally expensive from other models, the prediction graph does capture the downwards and upwards trend even though not matching exactly in the magnitude. Holiday and Weather data when included in the model gave slightly better results which indicate that weather and holiday features are important for the model's prediction and improve model performance. All the features are scaled to the range between 0 and 1 due to which the prediction results show mean, the standard deviation in the range of 0.25.

Feedforward networks are unable to capture the time series aspect of the data even though feed sequentially to the models. Binarized weather and holiday features degrade the model's prediction capability. Non-holiday data for training and holiday data for testing does not produce convincing results.

6.1.2 UNIVARIATE

After running all the RNN models 5 times each for both the combinations of the lag of 1 period and lag of 12 periods, it was evident that the model fed with data from the previous 12 periods as lag outperformed the model where data with only the last 1 period was used. Historic data helped the model to perform better and not be sensitive to any abrupt sudden change which can happen with last 1 period data. The model is able to make generalized predictions well. The RMSE was reduced to half for these models which were in the range of 330s whereas models with 1 lag period had RMSE in the range of 780s. Building complex RNN models with large numbers of layers and neurons were computationally expensive but the results didn't show substantial improvement to use this complex architecture.

GRU performed the best among all the models for both the combinations of lags used. Not only considering the average mean and standard deviation of RMSE and MSE was the deciding factor, but also the computational effort required was less for GRU having only 2 gates. The prediction graph shown in the results section shows that GRU is the closest to the actual traffic volume making it a clear winner when considering all the evaluation metrics for model performance comparison.The standard deviation of RMSE has reduced significantly for GRU when compared to LSTM and RNN.

CNN’s when compared to RNN models performed decently but are extremely computationally expensive 40 times more than GRU. Conv1D layer is used with filter size 64 and 2 such Conv1D layers are used for modeling. RMSE and MSE are also worse than any of the RNN models and the prediction graph shows that it performs well but RNN models are clear winners. CNN’s can be used for time series data when the time factor is not tightly coupled to the data as in this case it is.

6.1.3 MULTIVARIATE

Among all the models that we ran, we found that GRU had the best stats for prediction with the lowest RMSE and computational effort. Then, for the GRU model, we ran for lag = 12 and noticed that the RMSE was cut by half. This may be because there might be some values that were ignored and had greater values enough to affect the overall result. We tried for lag = 24 as well but found that there was an increase in RMSE. Thus, we settled for 12 (EXP1).

After that, we used the same hyperparameters from above with the same lag and ran models on a dataset with dropped columns which were weather and holiday (EXP2). This led to a change in RMSE. At first, it decreased then increased for next. This experiment showed that weather and holiday features are very important to the model’s prediction capability.

Thus for the final comparison, we took the mean and standard deviation of the RMSE for models after running 5 times and found that readings and graphs for EXP1 with weather and holiday data in it had more favorable results than EXP2.

By comparing Univariate time series analysis to multivariate time series analysis, it is seen that almost the same computational effort required RMSE and MSE more. But the effect of holiday and weather data has improved the prediction power of the model and the model is able to capture the nuances which are not present in the univariate time series.

6.2 FUTURE WORK

We would want to incorporate the spatial domain with the currently used temporal domain to make a combined Spatial-Temporal Traffic Volume Prediction which involves traffic images analyzed using CNN’s along with RNN models for temporal features to enhance the prediction capability of the deep neural networks (DNN).[3]

We want to use Encoders and Decoders for time series data modeling using RNN models in the future. Encoders and decoders are very useful when predicting the multistep time series data with the output not limited to just 1 at every interval but consist of mult- sequence prediction at every interval.[10]

APPENDIX

For the work done after final project implementation, we have implemented CNN for univariate time series data forecasting with a lag of 12 periods. The results obtained are included in the experiments section showing the results achieved by using CNN for univariate time series modeling. CNN’s were computationally extremely expensive as compared to RNN models to give a similar performance in terms of average RMSE. CNN’s with complex network configurations having more hidden layers or filters did not contribute to the model performance improvement. CNN’s have a high standard deviation for RMSE which shows that they are prone to high fluctuations when running the same model at different times.

For the project implementation submitted earlier, we had set the seed to generate reproducible results which was removed and all the models with their different architectures were re-run 5 times to see how the model's performance changes on RMSE with mean and standard deviation being calculated for comparing it with other models.

REFERENCES:

[1]https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

[2]Li, Wan, Wang, Jingxing, Fan, Rong, Zhang, Yiran, Guo, Qiangqiang, Siddique, Choudhury, and Ban, Xuegang. *Short-term traffic state prediction from latent structures: Accuracy vs. efficiency*. United States: N. p., 2019. Web. doi:10.1016/j.trc.2019.12.007.

[3]arXiv:1803.01254 [cs.LG]

[4]Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding, Transportation Research Part C: Emerging Technologies

[5]A. Koesdwiady, R. Soua and F. Karray, " Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508-9517, Dec. 2016, doi: 10.1109/TVT.2016.2585575.

[6]D. Zhang and M. R. Kabuka, " Combining weather condition data to predict traffic flow: A GRU based deep learning approach", *Proc. Int. Conf. Pervasive Intell. Comput. Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congre.*, pp. 1216-1219, 2018.

[7]https://jingqingz.github.io/files/projects/2017-09-EnglandHighway/report.pdf

[8]https://en.wikipedia.org/wiki/Long_short-term_memory

[9]https://en.wikipedia.org/wiki/Gated_recurrent_unit

[10]https://towardsdatascience.com/encoder-decoder-model-for-multistep-time-series-forecasting-using-pytorch-5d54c6af6e60

[11]Zeinab S Jalali, Chilukuri K. Mohan. LSTM Based Time-Series Link Prediction