

# Traffic Volume Prediction using Memory Based Recurrent Neural Networks

Semester project report for Data Mining, Dr. Xiaofei Zhang Instructing  
Lokesh Chandra Das, UID: U00740183

- Department of Computer Science, University of Memphis, Memphis, TN, 38111, ldas@memphis.edu

## Abstract

Predicting traffic volume in real time can improve both traffic flow and road safety. A precise traffic volume forecast helps alert drivers to the flow of traffic along their preferred routes, preventing potential deadlock situations. Existing parametric models cannot reliably forecast traffic volume in dynamic and complex traffic conditions. Therefore, in order to evaluate and forecast the traffic volume for every given time step in a real time manner, we develop non-linear memory-based deep neural network models. Our extensive experiments run on the Metro Interstate Traffic Volume dataset demonstrate the effectiveness of the proposed models in predicting traffic volume in highly dynamic and heterogeneous traffic environments.

## 1 Introduction

Rapid socioeconomic development aids in the growth of large-scale, expanding smart cities with easy access to communication technologies. Modern mobile and vehicular communication technology promotes Intelligent Transportation Systems (ITS), resulting in an exponential increase in the number of vehicles on the road each year. Traffic congestion has become one of the most pressing issues to address[1] and it is creating deadlock situations for large cities as well as for medium and small cities[2]. Traffic congestion can be mitigated using traditional methods by changing the road and urban infrastructures. However, redesigning the city structure to improve the traffic pattern and reduce congestion is very expensive, let alone time-consuming. Therefore, dynamic route planning, optimizing road allocations, managing traffic on urban roads, and using modern technologies to understand traffic patterns better are crucial tasks to reduce traffic congestion successfully. Forecasting the future traffic status based on past historical data is a way of reducing traffic congestion[3]. An accurate traffic volume prediction can alleviate traffic congestion and optimize traffic distributions.

Different techniques for traffic flow prediction have been studied lately. These methods can be generally classified as naive, parametric, and non-parametric. The naive model does not make any assumptions, and it is computationally fast. However, it has low accuracy. The parametric method includes different time-series methods. One of the most widely used parametric methods is the ARIMA (autoregressive integrated moving average) model[4]. The ARIMA model is practically effective in predicting traffic flow and has been a benchmark. However, the parametric approaches can achieve better performances if the time-series data shows a

regular pattern. In particular, the parametric methods fail to perform better when the traffic pattern varies in nature. Non-parametric methods address the issue of parametric methods, and various non-parametric methods such as, non-parametric regression, support vector machine, Kalman filtering, and neural network predictors are being used in traffic volume predictions. Deep neural networks have been shown to be superior at dealing with traffic forecasting. Recurrent neural networks (RNN), especially long-short Term memory (LSTM), demonstrated their advantages in modeling and predicting traffic flow.

In this project, we develop traffic volume prediction using long-short term memory (LSTM) and gated recurrent units (GRU) and evaluate our model on Metro inter-state traffic dataset[5].

## 2 Related Work

Traffic forecasting is critically important, and it has been extensively studied lately. Traffic prediction methods can be divided into parametric and non-parametric methods. Previously, researchers have applied various parametric methods to predict the traffic flow. The autoregressive integrated moving average is one of the most widely used parametric models in time-series prediction datasets. Chen et al.[6] predicted traffic flow using the autoregressive integrated moving average (ARIMA) model. However, other parametric models have also been used by the researchers. Kumar et al.[7] developed Kalman filtering techniques to forecast the traffic flow. Dong et al.[8] applied the gradient-boosting decision tree algorithm to perform short-term traffic flow prediction.

Deep learning-based approaches have recently received a lot of attention from researchers, and they provide more accurate estimation of traffic volume prediction than traditional parametric methods. The long-short term memory (LSTM) and gated recurrent unit (GRU) are capable of holding a long sequence of past observations and making a correlation in sequence prediction, which makes them more suitable for time-series datasets. Many researchers are using LSTM and GRU for predicting traffic conditions. Zhao et al.[9] used LSTM to predict the short-time traffic. Fu et al.[10] used LSTM and GRU for traffic flow prediction. Some researchers use a graph attention network to predict the traffic pattern[11]. In this project, we use LSTM and GRU to predict future traffic flow and evaluate the model using a metro interstate traffic volume dataset.

### 3 Problem statement

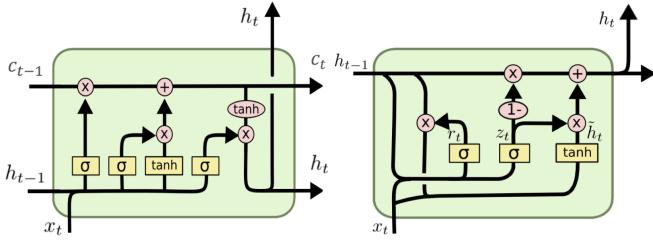
We consider the real-time traffic volume prediction as a multi-variate time-series problem where our model will approximately estimate future traffic flow based on the current and  $t$ -hours of historical observations. Specifically, our objective is to foretell future traffic volume at time step  $T_{t+1}, T_{t+2}, \dots, T_{t+f}$ , where  $f$  is the future prediction horizon using past observations from time steps  $T_{t-l}, T_{t-l-1}, \dots, T_{t-1}, T_t$  where  $l$  is the length of past observations used to predict the future traffic flow.

### 4 Method

We use memory-based recurrent neural network models, e.g., long-short term memory (LSTM) and gated recurrent units (GRU), to predict the future traffic flow.

#### 4.1 Long-Short Term Memory (LSTM)

Long-Short-term Memory networks (LSTMs)[12] are designed to learn long-term dependencies and effectively deal with the vanishing gradient problem of recurrent neural networks(RNNs)[13, 14]. Because it can hold long sequences while predicting the current output, LSTM is well suited for applications that make predictions based on time-series data. The memory block of the LSTM cell makes it easy to hold the sequence information. The memory block has memory cells and three gates: the forget gate, the input gate, and the output gate. The basic structure of an LSTM cell is depicted in Fig.1a.



(a) A structure of an LSTM cell (b) A structure of a GRU cell

The forget gate determines which information should propagate for the next time sequence through the sigmoid activation function. The input gate decides which information is necessary for the current state, and the output gate regulates what to output to the next state, e.g., the current output and the value of the next hidden state  $h_t$ . The equations to update the gates are as follows:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned}$$

where  $f_t$  is the forget gate,  $i_t$  is the input gate, and  $o_t$  is the output gate, respectively;  $C_t$  is a memory cell to hold the sequences from the previous states; and  $h_t$  is the output for the next state;  $W^*$  is weight, and  $b^*$  is bias.

#### 4.2 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) replaces the three gates of LSTM with two gates: the reset gate and the update gate. These gates use sigmoid activation functions linked by LSTMs, constraining their values between 0 and 1. Intuitively, the reset gate controls how much of the previous state we might still want to remember, and an update gate would allow us to control how much of the new state is just a copy of the old state. The Fig. 1b illustrates the inputs for both the reset and update gates in a GRU, given the input of the current time step and the hidden state of the previous time step. The mathematical mechanism for the GRU is as follows:

$$\begin{aligned} r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\ z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\ \tilde{H}_t &= \tanh(W_h \cdot [(h_{t-1} \odot r_t), x_t] + b_h) \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{H}_t \end{aligned}$$

where  $r_t$  is the reset gate,  $z_t$  is the update gate, and  $h_t$  is the current time step's output. The rest of the symbols contain the same meaning as described in the LSTM section.

#### 4.3 Dataset

We use Metro Interstate Traffic Volume Data Set[5]. This is a multivariate, sequential, time-series dataset collected from the westbound I-94 at Minneapolis-St. Paul, Minnesota. The dataset is relatively large and has 48204 instances with nine features. The dataset collected traffic volume data in an hourly manner from 2012 to 2018, taking into account weather features and holidays that impact the traffic volume.

#### 4.4 Data Preprocessing

We cannot directly feed data into the LSTM or GRU models, unlike other deep learning models such as CNN and RNN. We have to convert it into a specific format. Input format should have at least time steps, and number of features. Generally, in time series prediction, we use  $t$ -hours of observations as input to the network, and the model will produce output at  $t+1$  hour. Here, we use the past  $t$ -hours of data to predict the next  $n$  hours' traffic volume. The dataset also contains some categorical values that needed to be converted into numerical values. Moreover, the attributes are in different scales. The statistics of the dataset are shown in Fig.2. It is clear that traffic\_volume attribute values are very larger compared to rain\_1h values. So we need to scale them to reduce the bias. We use the MinMaxScaler technique to normalize the feature values between 0 and 1. We split the dataset into training and testing sets. We use data from year 2012-2017 for training among which 20% was used for validation purpose and last one year data was used for testing purpose. From the ex-

	holiday	temp	rain_1h	snow_1h	clouds_all	weather_main	weather_description	traffic_volume
count	47284.0	47284.000000	47284.000000	47284.000000	47284.000000	47284.000000	47284.000000	47284.000000
mean	1.0	280.839466	0.132735	0.000227	49.790986	2.605173	16.564631	3232.676233
std	0.0	12.451778	1.012858	0.008247	39.064691	2.787757	8.916584	1989.588639
min	1.0	243.390000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.0	272.020000	0.000000	0.000000	1.000000	0.000000	10.000000	1151.000000
50%	1.0	281.930000	0.000000	0.000000	64.000000	1.000000	17.000000	3319.000000
75%	1.0	291.450000	0.000000	0.000000	90.000000	5.000000	27.000000	4919.250000
max	1.0	301.620000	55.630000	0.510000	100.000000	10.000000	37.000000	7280.000000

Figure 2: Statistics of the dataset

tensive data analysis, we found that some features contained

outliers. For example, *rain\_1h* and *temp* features have outliers. We remove them before feeding them into the model to improve its generalization capability. We use the interquartile range technique to remove the outlier. Everything that is 1.5 times less than the first interquartile range or 1.5 times greater than the last interquartile range is removed. Figures in Fig. 3 show the results.

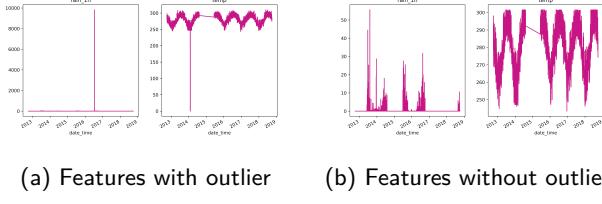


Figure 3: Dataset contains outliers

## 5 Experimental Setup

We run a total of 4 experiments with different settings, taking 6 to 24 hours of past observations to predict the next hour's traffic volume. We also did an experiment to see how the features impacted the traffic volume prediction. In one set of experiments, we select all features from the dataset, and in another set of experiments, we only consider four features, namely *temperature*, *rain\_1h*, *clouds\_all*, and *traffic volume*. We run experiments on two sets of neural networks. The table 1 shows the hyperparameters for our deep learning model. For the experiment 2, we just changed the hidden number of units as like [256, 128, 64, 32] and number of epochs is 500. The rest of the hyperparameters are the same as those shown in the table. The training is stopped if the

Table 1: Table

Neural Network Settings	
Input Shape	(# of Records, Time Step, # of Features)
Hidden Layers	4 Hidden Layers with 128, 64, 32, 16 units of neurons in each layer respectively
Activate Function	tanh
Batch Size	64
Learning Rate	0.0001 with decay rate 1e-5
Optimizer	Adam
Epochs	300

validation error does not improve for at least five consecutive runs. The main evaluation metrics are mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) as shown in equations 1, 2, and 3.

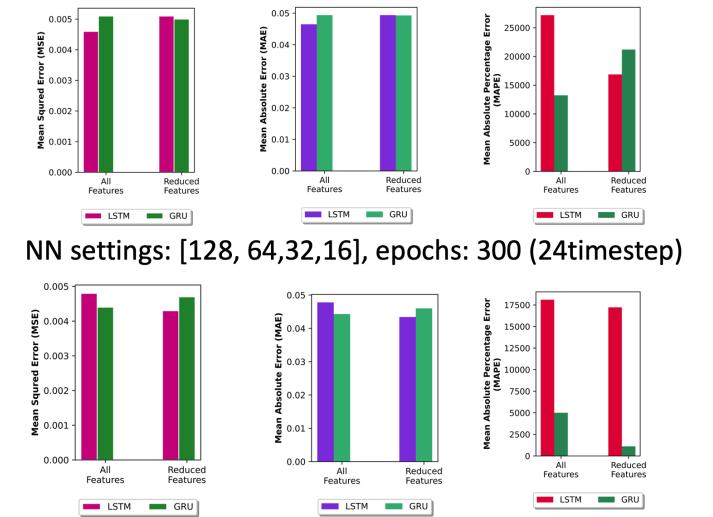
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

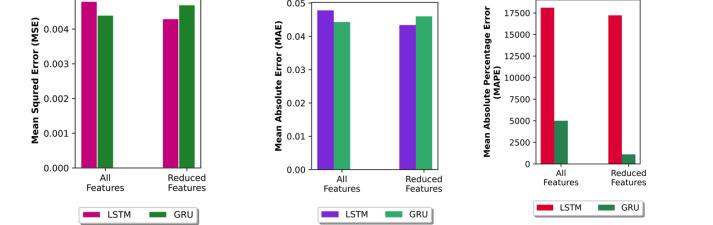
$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (3)$$

## 6 Results

We implemented the traffic volume prediction problem in Python based on Keras and Tensorflow. To train and test the proposed model, a workstation equipped with an Intel Xeon Gold 5222 processor, an NVIDIA® RTXTM A4000 graphics card, and 48GB of RAM running Windows 11 OS is used. Empirically, the autoregressive integrated moving average (ARIMA) is not suitable when the dataset is more complex and has long sequences. So, we directly choose LSTM and GRU, and ARIMA is out of the scope of the project. In our experiment, we compare the LSTM and GRU performances by varying the neural network settings and also by varying the length of historical observations. We compare the mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) for both GRU and LSTM with different neural network settings, taking into account all features vs. a reduced number of features. Fig. 4 shows the



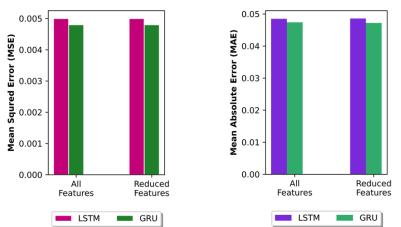
NN settings: [128, 64, 32, 16], epochs: 300 (24timestep)



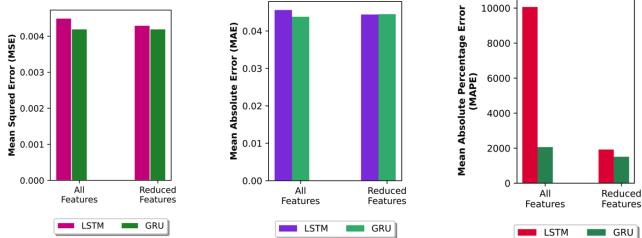
NN settings: [256, 128, 64, 32], epochs: 500 (24timestep)

Figure 4: Evaluation metrics for 24 past observations

comparative results of two models (e.g., LSTM and GRU) for different feature sets and neural network settings when the length of past observation is 24. Although there is no significant pattern in the results, it can be seen that both the GRU and LSTM models produce better results in terms of MSE and MAE. On the other hand, when training does not require preserving a very long history, GRU performs better than LSTM. Fig.5 shows the results when the length of the past observation is 6. It can be observed that when the past observation history is very limited, GRU performs better compared with the LSTM in all three evaluation metrics for both neural network settings. Fig.6 and Fig.7 depict the predicted traffic volume results for various times. Here, the blue line indicates the actual ground truth, and the orange line indicates the predicted value. Both LSTM and GRU can predict the traffic volume as close to the actual values as possible. However, GRU outperforms LSTM when the past observation sequences are small, and LSTM performs better when the dataset is complex and requires the use of very long sequences to predict the future traffic volume. Fig.8 shows the convergence of the LSTM and GRU models. Both models take more training time to get convergent when the



NN settings: [128, 64, 32, 16], epochs: 300 (6 timestep)



NN settings: [256, 128, 64, 32], epochs: 500 (6 timestep)

Figure 5: Evaluation metrics for 24 past observations

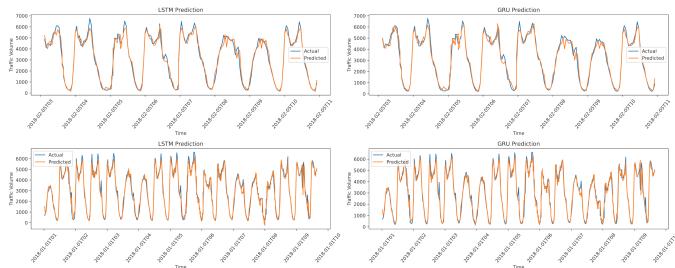


Figure 6: Traffic Volume Prediction with 6 past observations

number of features is less. However, GRU takes more time compared to the LSTM.

## 7 Deliverable

The [Traffic Volume Prediction](#) repository contains the implementation, dataset, and trained models. Instructions on how to run can be found in the README.md file.

## 8 Discussion

In this section, I will discuss challenges, limitation, computational efficiency, and potential other methods for predicting traffic volume.

### 8.1 Challenges and Limitations

Preparing the dataset to feed into the LSTM and GRU networks was one of the more challenging tasks as it added another dimension (e.g., time) to the input shape. Moreover, to find a proper set of hyperparameters, the network required a significant amount of time, which was very challenging given the limited amount of computational resources. Finally, we have used MinMaxScaler to normalize the dataset. Whenever we compared the results with ground truth, we had to convert it back to its original format, which also requires a lot of work.

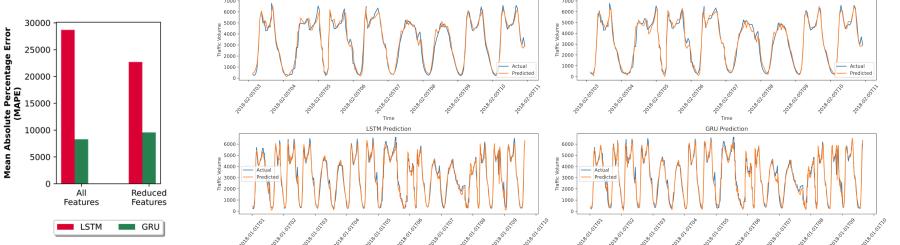


Figure 7: Traffic Volume Prediction with 24 past observations

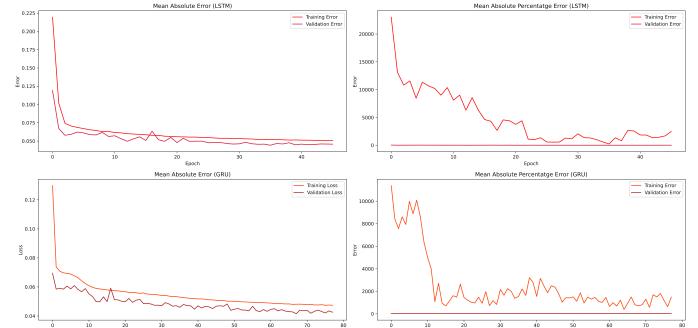


Figure 8: Training convergences for both LSTM and GRU

## 8.2 Computational Time and Memory

LSTM and GRU are computationally expensive and require a lot of memory, as they use memory to store the historical observations internally. To finish a sing epoch training given the above network structures and hyperparameters settings, both network took approximately  $25 \sim 28s$ .

## 8.3 Other methods

Recently, many researchers used graph-based attention modules to predict time series data. Graph-based multi-attention networks could be applied to predicting the traffic volume as well.

## 9 Conclusion

In this project, we develop memory-based deep recurrent neural network models, namely LSTM and GRU, to predict the traffic volume. The models are evaluated on the widely used Metro Interstate Traffic Volume dataset. The experimental results demonstrate the effectiveness of the proposed LSTM and GRU models. To obtain better predictions, however, complete hyperparameter tuning and extensive experiments are required, which are our future works.

## References

- [1] Y. Fan, B. Cai, W. Shangguan, J. Wang, and D. Jiang, "Multi-objective operation control of rail vehicles," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1265–1270, IEEE, 2013.
- [2] L. Kuang, C. Hua, J. Wu, Y. Yin, and H. Gao, "Traffic volume prediction based on multi-sources gps trajectory data by temporal convolutional network," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1405–1417, 2020.

- [3] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proceedings of the 2017 SIAM international Conference on Data Mining*, pp. 777–785, SIAM, 2017.
- [4] C. Chatfield, *Time-series forecasting*. Chapman and Hall/CRC, 2000.
- [5] J. Hogue, "Metro interstate traffic volume data set," URL: <https://archive.ics.uci.edu/ml/datasets/Metro+ Interstate+ Traffic+ Volume>, 2019.
- [6] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 607–612, IEEE, 2011.
- [7] S. V. Kumar, "Traffic flow prediction using kalman filtering technique," *Procedia Engineering*, vol. 187, pp. 582–587, 2017.
- [8] X. Dong, T. Lei, S. Jin, and Z. Hou, "Short-term traffic flow prediction based on xgboost," in *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 854–859, IEEE, 2018.
- [9] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
- [10] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, 2016.
- [11] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1234–1241, 2020.
- [12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [13] Q. Li, F. Wang, J. Wang, and W. Li, "Lstm-based sql injection detection method for intelligent transportation system," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4182–4191, 2019.
- [14] L. Hou, J. Zhu, J. Kwok, F. Gao, T. Qin, and T.-y. Liu, "Normalization helps training of quantized lstm," *Advances in Neural Information Processing Systems*, vol. 32, 2019.