Journal of Innovative Engineering and Research (JIER)
Vol.- 7, Issue -1, April 2024, pp. 21-25 (5 pages)

Performance Evaluation for Traffic Flow forecasting using LSTM with Different Time Scales

MOHIT KUMAR VARMA, DR. GAURAV GUPTA

Department of Computer Application
Dr. A. P. J. Abdul Kalam University, Indore 452010, India
Correspond Author Email: mohitansh12@gmail.com

Abstract— *One of the key difficulties in traffic management and system guidance has been traffic congestion in smart and big cities. The first problem, owing to rapid economic expansion and an ever-increasing number of cars, is to accurately estimate traffic flow information in order to reduce traffic congestion and accidents. Many academics have recently begun to concentrate their efforts on deep learning approaches, such as Recurrent Neural Networks (RNN), because of their ability to learn long-term dependencies of sequence data and capture the non - linearity characteristic of traffic flow. By considering different time intervals, this article employed three distinct types of recurrent neural network architecture, such as basic RNN, Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU). The information gathered from the California Department of Transportation in 2018 and 2019 was used; however, owing to inaccurate measurement and equipment problems, a few missing values were detected. The mean approach on the same hours was used to calculate and substitute the missing values in this study to assure the data quality to be trained in our models and boost model performance. In this paper, the LSTM model is suggested for both short and long time periods. The prediction efficiency was assessed using two prominent metrics: Mean Absolute Percentage Errors (MAPE) and Root Mean Squared Error (RMSE).*

Index terms: *LSTM*

I. INTRODUCTION

Because of the growing number of automobiles, traffic congestion and statistics in modern regions have exploded in recent years. Individual passengers and the Intelligent Transportation System (ITS) specify that traffic flow is vital for both drivers since they are stuck in traffic for long periods of time. Electronic devices are being used to capture traffic data such as passing vehicle statistics, including quantities, speed, and class at a certain time [1]. However, the extensive examined data acquired may be used to assist transportation planners in improving current road networks or building new ones based on expected long- and short-term traffic flow. All of these are based on ITS [2, 3], which is the fundamental traffic forecast system. Although exact traffic prediction is a difficult topic to solve, the large traffic data gathered contains missing or wrong information due to a variety of factors such as equipment malfunctions and imprecise measurement, resulting in inaccurate predictions and low quality output. Data preparation, in which the dataset is prepared and cleansed, is one of the greatest remedies to such flaws [4]. Traffic forecasting techniques have gradually transitioned from statistical models to machine learning intelligence, and are divided into two categories: parametric and non-parametric models [5–7]. Furthermore, due to the stochastic and nonlinear characteristics of traffic flow, parametric linearity methods did not provide high efficiency in predicting future situations, and more researchers began to focus on non-parametric methods that attempt to learn historical data that is related to the expectation instant and use the information items found to forecast for the future.

Many traffic flow forecasting methodologies have been developed, with a focus on short-term traffic flow prediction, which is still a difficulty today [8]. The most widely employed approach in parametric models, according to the literature, is the Autoregressive Integrated Moving Average (ARIMA), which assumes that the traffic state is stationary. One of its flaws is ARIMA's natural proclivity to focus on the data's mean values from the previous series. As a result, capturing a quickly shifting phase remains challenging [9]. As a result of the failure of nonlinear and stochastic parametric models to forecast effectively, more researchers have explored and created non-parametric models, including the Support Vector Regression (SVR) application successfully presented for the prediction of time series. It has shown significant flaws, such as the lack of defined methods for determining some key model parameters [9]. In the realm of traffic studies, the use of neural networks has sparked a lot of attention.

The distinction between standard models and neural networks clearly shows an advantage in forecasting correct traffic information [10]. Recurrent Neural Network (RNN) is one of the deep learning models that has established a reputation for dealing with time series via recurrent neural ties; however, Gers et al. in [11,12] show that there are still many problems to be addressed in fashion because RNNs do not train with long time lags in the time series, despite the fact that this incident is commonly seen in traffic prediction tasks. Second, RNNs rely on predefined time delays to learn the processing of temporal series, but it is difficult to find the appropriate time window size in an automated manner. Long Short Term Memory (LSTM) has been created to tackle the problem by changing the placement of the secret neurons in a regular RNN. Wang et al. [13] use an LSTM-based technique to estimate traffic load in a certain geometric field for the following instant. LSTM was used to forecast traffic speed using data from distant microwave sensors in [14]. Yongxue Tian and Li Pan [8] evaluated several non-parameter models, including SVM, SAE, FFNN, and LSTM RNN, and concluded that the LSTM RNN model delivers the best results. To forecast traffic flow, Li et al. evaluated the LSTM and GRU models [15].

In this research, we examine and propose the LSTM model, which has been compared to GRU and Simple RNN, which are all known to have similar RNN architectures. The best model for short-term traffic prediction is compared to four distinct time frame portions of 1 hour to 4 hours expected future results. The remainder of this work is structured as follows: Section II describes the LSTM; Section III shows the experimental setup; Section IV shows the findings; and Section V summarises the conclusion and future study.

II. DESCRIPTION OF LSTM

LSTM Overview

LSTM is the most robust and well-known subclass of RNN. Both are artificial neural networks that are trained to identify patterns in data sequences such as numerical time series data, stock exchanges, and government organisations. LSTM is a type of RNN that can learn long-term dependencies; the key principle underlying the LSTM design is a memory cell that can keep its condition over a long period of time and regulates the transfer of information out of the cell. The conventional LSTM has one input layer, one hidden repeating layer, one layer with a memory block as the fundamental unit, and one output layer. The memory block is made up of three adaptive, multiplicative memory cells and a self-connected memory cell with temporal state memorization. Input, output, and gates to control the forgotten gates are all examples of gating devices. The flow of data within the block. The three extra gates enable access to the block's Continuous equivalents of write, read, and reset operations. Multiplicative gates may learn to open and close, allowing LSTM memory cells to retain and retrieve information over lengthy periods of time. Resolving the vanishing gradient problem.

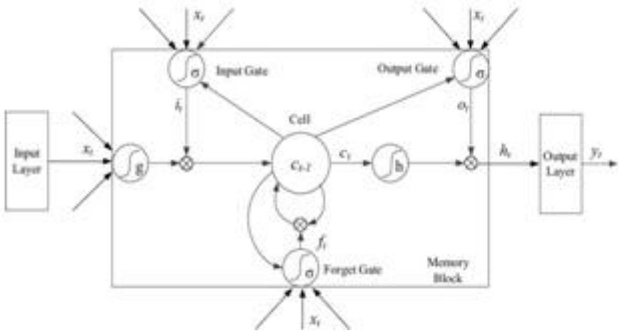

Figure 1 shows an example of an LSTM memory block.

III. RESULT AND ANALYSIS

The historical traffic data is denoted as x= (x1, x2, ….,xT), the hidden state of the memory cell is represented as h= (h1,h2,….,hT) and the traffic data predicted as y=(y1,y2,….,YT).The networks of LSTM do the computations as follows:

The W term denotes weight matrices (e.g. W_{xh} is the input-hidden weight matrix), b term denote bias vectors (e.g.) b_h is hidden bias vector). H is the hidden layer function, which is performed by the following composite function:

$$h_t = H(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy} h_t + b_y \quad (2)$$

$$i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t g(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o) \quad (6)$$

$$H_t = O\, h(c_t) \quad (7)$$

A. *Data Description and Experimental Design*

The dataset used in this study was obtained from the performance measuring system of the California Department of Transportation (Caltrans) (PeMS). It is one of the most widely utilised databases in the field of traffic flow data. We used data from individual sensors on the SR237-W highway system in Santa Clara County, California, which is located in the city of Sunnyvale. The data was gathered from the 1st of January 2018 to the 31st of December 2019 with an update frequency of 30s and then aggregated for each detector station into 5min, 1hour, daily, or weekly intervals. The total number of sample points in the dataset we utilised was 12000, with 80 percent used for training and the remaining 20 percent used for testing. The raw dataset utilised in our experiment is separated into 1-hour intervals each day. The traffic flow data, on the other hand, appears to have a one-day cycle constituted of 24-hours, with workday patterns considerably distinct from vacation and weekend patterns. As observed in the literature, the trend of removing weekends is highly widespread [16-18], and in this article, working days are the only ones chosen, as shown in Fig 2, and the two peak hours are 8 AM and 5 PM, when about 2500 cars may pass in just one hour.

All of the algorithms in our experiment were written in Python, with TensorFlow as the backend and the Keras library. To get accurate prediction results, we created our model using the following methods, which focus on deep learning data preprocessing:

- Step 1: The first and most crucial step is to get a relevant and up-to-date dataset from the California Department of Transportation (Caltrans) performance monitoring system (PeMS).

Journal of Innovative Engineering and Research (JIER)
Vol.- 7, Issue -1, April 2024, pp. 21-25 (5 pages)

• Step 2: We imported all of the essential Python libraries, as well as the dataset.

• Stage 3: Preprocessing to detect and replace the missing values in our dataset. The missing records have been replaced by the historical average mean value of the same prior hour to assure an accurate result.

• Step 4: Normalize the data by scaling within a range of 0 to 1 while training and applying data analysis. The comparison of statistics will be challenging if the data is vast.

• Step 5: Split the complete dataset in 80% and 20% ratio for training and test set respectively for determining the model input and output values.

• Step 6: Create a model by defining all of the parameters, such as the number of layers and neurons.

• Step 7: Now, LSTM model can be trained, and the results will be examined before the parameters are changed.

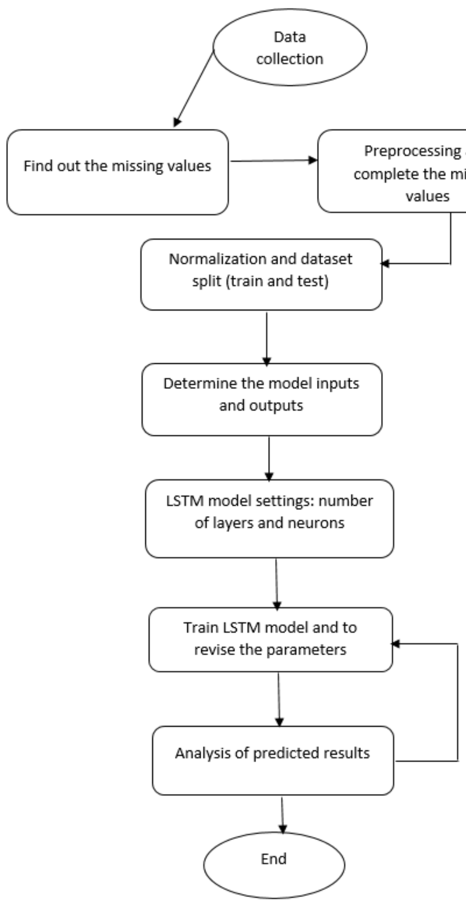

Figure 2 Flowchart of short term traffic flow prediction based of LSTM

For our experiment, we solely take traffic flow data as the prediction input, without taking into account additional factors such as road accident data, atmospheric conditions, or other fundamental traffic flow metrics such as speed and density. The following table I details some of the primary optimal parameters of the proposed model in short term traffic prediction, including the size of the input layer, the number of hidden layers and the hidden units in each hidden layer, the number of epochs, the activation function, the batch size, and the output layer size. Modifying the settings

B. Index of Performance

In this research two popular metrics have been used to evaluate the accuracy of the short term traffic flow, including both Root Mean Squared Error (RMSE) which is a common way to calculate a model's Error in quantitative data prediction, and Mean Absolute Percentage Errors (MAPE) which tests the prediction accuracy of a forecasting system usually presented in percentage.

RESULTS:

In terms of predicting accuracy and dependability, the findings demonstrate that most of the variances are detected quite well. A comparative performance study of three forecasting models, including Simple RNN, GRU, and LSTM, is provided, and several useful conclusions are interpreted:

1. The short-term traffic flow forecast effectiveness of all three distinct models rises as the time interval grows from one hour to the fourth hour.

2. Both GRU and LSTM have demonstrated to be more accurate than Simple RNN as the best model for predicting short-term traffic flow. When prediction stability is taken into account, however, LSTM beats GRU.

3. The LSTM makes use of its ability to continually update the input stored in its memory. During training, this allows the model to remember the pattern, trend, and fluctuation in the dataset for a long period in memory.

The comparison between actual and expected car per hour traffic flow numbers is shown in our model attempted to represent the true values, as seen by the findings, which show some connections. Because the peak hours on our graph are seven a.m. and four p.m., traffic flow is very low from midnight (zero hour) to the fourth hour early in the morning and from ten to eleven p.m., implying that the number of cars cannot influence the traffic flow.

To compare and validate the efficiency of the proposed model LSTM, other types of RNN prediction models, such as GRU and Simple RNN, were chosen. Cho et al. (2014) proposed that GRU features gating units that regulate data flow inside the unit, whereas Simple RNN and LSTM compute a weighted sum of the inputs and apply tanh as a nonlinear function [19]. The architecture and prediction model procedure of all of the prediction models chosen are identical. The average outcomes of the three prediction models' RMSE and MAPE values are reported in table II and table III, according to a different time interval.

Predicted time	Models		
	Simple RNN	GRU	LSTM
1-Hour	1 5 9.7 9	154.66	149.52
2-Hours	308.98	304.77	314.41
3-Hours	469.29	463.89	436.92
4-Hours	583.89	592.25	584.63

TABLE I. Prediction performance(RMSE)

Table II shows that GRU computed a modest difference between observed and anticipated error levels for the second hour, based on the RMSE values. The two metrics RMSE and MAPE are near in all RNN architecture models' prediction performance, especially in table III, where the MAPE of LSTM and GRU is 11.04 percent and 11.70 percent, respectively. The fourth hour's traffic flow estimate beat all of the others. As the proportion decreases, using prior data in the models to improve forecast accuracy and re-train to the following hour may be beneficial. As a result, LSTM and GRU may learn and remember long-term dependencies.

Predicted time	Models		
	Simple RNN	GRU	LSTM
1-Hour	15.6 1 %	14.60%	13.28%
2-Hours	15.59%	14.72%	12.92%
3-Hours	15.09%	13.04%	12.03%
4-Hours	13.76%	11.70%	11.04%

TABLE II. Prediction performance(MAPE)

V. CONCLUSION

In this study, three distinct RNN architectures were used to forecast traffic flow across short and long time intervals, including Simple RNN, GRU, and LSTM. To assure the pre processed data quality, a few missing values in the dataset gathered on Caltrans PEMS in 2018 and 2019 were substituted using the same missing hour's mean approach. In our research, LSTM outperformed GRU and RNN. However, GRU has shown results that are more similar to our suggested technique, particularly in the fourth-hour traffic flow forecast, where LSTM's MAPE is 11.04 percent and GRU's is 11.70 percent. The sole input used in this research was traffic flow. In future studies, other elements such as vehicle speed and weather conditions will be addressed to improve the RNN models prediction performance.

REFERENCES

[1]. M. Jiber, I. Lamouik, Y. Ali, and M. A. Sabri, "Traffic flow prediction using neural network," 2018 Int. Conf. Intell. Syst. Comput. Vision, ISCV, 2018, vol. 2018-May, pp. 1–4, 2018, doi: 10.1109/ISACV.2018.8354066.

[2] X. Chen and R. Chen, "A Review on Traffic Prediction Methods for Intelligent Transportation System in Smart Cities," Proc. - 2019 12th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI, 2019, no. 5,2019,doi:10.1109/CISPBMEI48845.2019.8965742.

[3] E. Bolshinsky and R. Freidman, "Traffic Flow Forecast Survey," Tech. Inst. Technol. Report.–15,pp.1–15,2012,[Online].Available: http://nwwwn.cs.technion.ac.il/users/wwwb/CGI-bin/tr-get.cgi/2012/CS/CS-2012-06.pdf.

[4] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," Appl. Artif. Intell., vol. 17, no. 5–6, pp. 375–381, 2003, doi: 10.1080/713827180.

[5] S. Oh, Y. J. Byon, K. Jang, and H. Yeo, "Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach," Transp. Rev., vol. 35, no. 1, pp. 4–32, 2015, doi: 10.1080/01441647.2014.992496.

[6] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," Transp. Res. Part C Emerg. Technol., vol. 43, no. February 2018, pp. 3–19, 2014, doi: 10.1016/j.trc.2014.01.005.

[7] L. Lin, Q. Wang, and A. Sadek, "Short-term forecasting of traffic volume," Transp. Res. Rec., vol. c. No. 2392, pp. 40–47, 2013, doi: 10.3141/2392-05.

[8] Y. Tian and L. Pan, "Predicting Short-term Traffic Flow by Long Short-Term Memory Recurrent Neural Network," 2015, doi: 10.1109/SmartCity.2015.63.

[9] W. Hong, "Application of seasonal SVR with a chaotic immune algorithm in traffic flow forecasting," pp. 583–593, 2012, doi: 10.1007/s00521-010-0456-7.

[10] P. Poonia and V. K. Jain, "Short-Term Traffic Flow Prediction: Using LSTM," Proc. - 2020 Int. Conf. Emerg. Trends Commun. Control Comput. ICONC3 2020, 2020, doi: 10.1109/ICONC345789.2020.9117329.

[11] T. Ese N et al., "Long Short-Term Memory in Recurrent Neural Networks," vol. 2366, 2366.

[12] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Comput., vol. 12, no. 10, pp. 2451–2471, 2000, doi: 10.1162/089976600300015015.

[13] J. Wang et al., "Spatiotemporal modelling and prediction in cellular networks: A big data-enabled deep learning approach," Proc. - IEEE INFOCOM, 2017, doi: 10.1109/INFOCOM.2017.8057090.

[14] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," Transp. Res. Part C Emerg. Technol., vol. 54, pp. 187–197, 2015, doi: 10.1016/j.trc.2015.03.014.

[15] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction," no. November 2016, 2018, doi: 10.1109/YAC.2016.7804912.

[16] Y. Kamarianakis and P. Prastacos, "Space-time modelling of traffic flow," Comput. Geosci., vol. 31, no. 2, pp. 119–133, 2005, doi: 10.1016/j.cageo.2004.05.012.

[17] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2, pp. 871–882, 2013, doi: 10.1109/TITS.2013.2247040.

[18] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic Flow Prediction with Big Data: A Deep Learning Approach," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, pp. 865–873, 2015, doi: 10.1109/TITS.2014.2345663.

[19] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," pp. 1–9, 2014, [Online]. Available: http://arxiv.org/abs/1412.3555.