Short-Term Traffic Forecasting using LSTM-based Deep Learning Models

Dilantha Haputhanthri
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
dilantha.15@cse.mrt.ac.lk

Adeesha Wijayasiri
Department of Computer Science and Engineering
University of Moratuwa
Sri Lanka
adeeshaw@cse.mrt.ac.lk

Abstract—**Accurate short-term traffic volume forecasting has become a component with growing importance in traffic management in intelligent transportation systems (ITS). A significant amount of related works on short-term traffic forecasting has been proposed based on traditional learning approaches, and deep learning-based approaches have also made significant strides in recent years. In this paper, we explore several deep learning models that are based on long-short term memory (LSTM) networks to automatically extract inherent features of traffic volume data for forecasting. A simple LSTM model, LSTM encoder-decoder model, CNN-LSTM model and a Conv-LSTM model were designed and evaluated using a real-world traffic volume dataset for multiple prediction horizons. Finally, the experimental results are analyzed, and the Conv-LSTM model produced the best performance with a MAPE of 9.03% for the prediction horizon of 15 minutes. Also, the paper discusses the behavior of the models with the traffic volume anomalies due to the Covid-19 pandemic.**

Keywords—CNN-LSTM, Conv-LSTM, encoder-decoder, LSTM, traffic volume forecasting

I. INTRODUCTION

Short-term traffic forecasting has become a major asset in the field of real-time traffic management and transportation planning. Traffic volume forecasting can be utilized in optimizing routing strategies and journey planning to reduce traffic congestion and accidents [1]. Traffic is characterized by the volume of the vehicles passing through in a given time interval. Traffic data usually have a strong seasonality of 24 hours and also depends on the day of the week, and other cultural affiliations such as holidays. Because of this complexity of the data, univariate time series forecasting models and other classical statistical models are inadequate for modelling non-stationary traffic in complex road structures [2].

Most of the related work has been based on time series analysis and forecasting techniques such as ARIMA modelling for traffic forecasting [3]. However, these models only consider the temporal variations of the traffic, to capture the stochastic and nonlinear nature of the traffic, researchers have been compelled to explore machine learning techniques such as k-nearest neighbor (KNN), support vector regression (SVR) and artificial neural networks (ANN). However, in recent years, with the arrival of the big data era, deep learning has been performing well in the domain and several recent works on short-term deep learning-based traffic forecasting also have shown great strides in performance. Several types of deep learning models have been proposed for traffic forecasting [4]–[8]

Long-short Term Memory (LSTM) is one of the commonly used deep learning techniques, designed specifically for time series forecasting, by employing memory states and feedback connections, unlike standard feedforward networks.

In this paper, we propose LSTM based deep learning models with different architectures for understanding the best forecasting model. A simple LSTM model, encoder-decoder LSTM model, convolutional neural network LSTM (CNN-LSTM) model, and a Convolutional LSTM (Conv-LSTM) model are designed and developed. Then using a real-world traffic volume dataset, we evaluate the performance of the proposed models employing several error metrics namely, mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE). Finally, we will analyze and compare the obtained results. Nevertheless, the results are only compared among the proposed models and the prediction horizons, because of the absence of short-term traffic forecasting research done using the exact dataset. Also, the dataset contains the data during the Covid-19 pandemic period where a clear anomaly in traffic volume is observed. Proposed models are tested and optimized on those data and the results are discussed. This can be considered a novelty in the research domain.

The paper is organized as follows. Section 2 explores the background of the deep learning techniques used and the existing literature, section 3 describes the methodology, section 4 discusses the experimental results and the performance of the models during the Covid-19 pandemic and finally, section 5 concludes the paper.

II. BACKGROUND

A. Learning Algorithms

LSTM is a deep learning technique classified as a type of artificial recurrent neural network (RNN). RNNs are specifically designed for processing sequential data using the internal memory states and also support feedback connections unlike standard feedforward networks [9]. Fig. 1 illustrates a common LSTM neuron or a unit that is composed of a cell, input gate, output gate and forget gate. The cell memorizes the feedback values over time intervals and the gates control the information flow in and out of the cell. Each of these gates can be considered as standard neurons in a feedforward network in which the activation of a weighted sum is calculated. The calculated activations are i_t, o_t, and f_t for the input gate, output gate, and forget gate, respectively. The activations of the gates, f_t, i_t and o_t are calculated for a timestep t using the activation of the cell at t-1 (c_{t-1}) according to the equations (1), (2), and (3).

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (1)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

Here, the W and U represents the weights of the input and the recurrent connections of the input gate (i), output gate (o), forget gate (f) or the memory cell (c).

2021 Moratuwa Engineering Research Conference (MERCon) | 978-1-6654-3753-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/MERCON52712.2021.9525670

978-1-6654-3753-0/21/$31.00 ©2021 IEEE

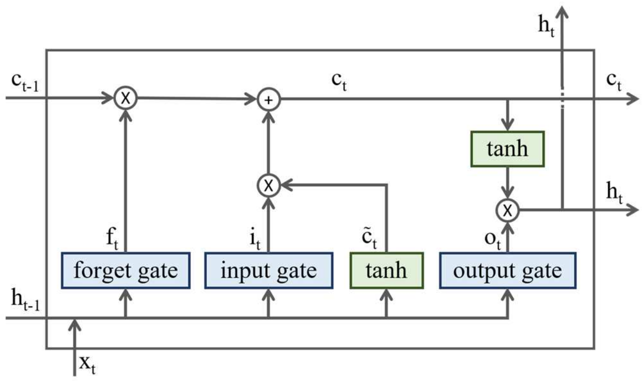

Fig 1. Structure of an LSTM unit

Finally, c_t and h_t are calculated using equations (4), (5) and (6).

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (5)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (6)$$

Here, σ represents the sigmoid function and in both Fig 1 and the equations, ⊗ denotes the element-wise multiplication. Furthermore, the upward output, h_t usually goes through another activation function, which is then considered as the output of the unit [10].

There are several variations to the LSTM model and three encoder-decoder models are used in this research namely, standard encoder-decoder LSTM, CNN-LSTM and Conv-LSTM. Encoder-decoder LSTM is a neural network architecture specifically designed to address sequence-to-sequence problems which often covers forecasting one to one or many to one sequence prediction scenarios. Here the model comprises two LSTM models that act as encoder and decoder. The encoder maps the variable length input sequence to a fixed-length vector while the other decodes the vector to a target sequence [11]. Encoder-decoder LSTM has been used in several recent related works in the traffic forecasting domain [8], [12]. In the CNN-LSTM model, a convolutional neural network (CNN) is used as the encoder which is capable of automatically deriving the important features for the learning process. Although CNN is supported for data with a 3D structure, multiple features can be configured to act like different channels, ultimately creating the same effect. An LSTM decoder is used as in the normal encoder-decoder model. There are several short-term traffic forecasting work done using CNN-LSTM models [6]. Conv-LSTM is a further extension of the CNN-LSTM where the convolutions are performed for each time step as a part of the LSTM. Conv-LSTM is also a learning model that has been often used in short-term traffic forecasting in recent years [4], [7].

B. *Related Work*

This section discusses some of the recent work done related to LSTM based traffic forecasting.

Haifeng Zheng et al. [4] proposed a hybrid deep learning approach using attention-based Conv-LSTM networks for short term traffic flow prediction, automatically extracting inherent features of traffic flow data. The main objective is to efficiently capture the complex nonlinearity of traffic flow increasing the prediction accuracy. The attention mechanism is designed to distinguish the importance of flow sequences at different times by automatically assigning different weights. Also, a bidirectional LSTM (Bi-LSTM) module was used to extract periodic features to identify variance tendency of the traffic flow from both previous and posterior directions, further exploring long-term temporal features. Performance analysis of the proposed models was done using a real-world dataset that contains both highway and urban traffic flow data. The analysis was carried out for several prediction horizons and the results show that the proposed model combining the attention-based Conv-LSTM and Bi-LSTM produce better performance than existing approaches for all the prediction horizons.

Zheng Zhao et al. [5] proposed a novel LSTM model for traffic forecasting which has a unique design with many memory units to capture the temporal-spatial correlation in a traffic system via a two-dimensional network. The proposed model produced better performance compared to the other representative forecast models. The used dataset was obtained by the Beijing Traffic Management Bureau from over 500 stations with a frequency of 5 minutes. The dataset contained features such as vehicle volume, lane occupancy and average velocity for more than 25 million validated records. Data that belong to three stations were selected where traffic volume is high, medium, and low. MAE, MSE and mean relative error (MRE) was employed as the evaluation metrics and the model produced 6.41%, 6.05%, and 6.21% MRE values for the selected stations concluding that the proposed model is effective and reliable for traffic forecasting.

A graph CNN-LSTM model was proposed by Toon Bogaerts et al. [6] for both short and long term traffic forecasting based on trajectory data. A sparse trajectory (GPS) dataset that was produced by the ride-hailing service of DiDi in two cities of China, was used for the evaluation. Along with the learning model, the authors also propose a temporal correlation-based data reduction method in selecting the most relevant road links as the model inputs. The combination of this method and the learning model produced better results compared to the other work that had been done in the domain such as standard LSTM. Several different prediction horizons from 5 min to 4 hours were tested and the performance of the proposed model was compared with several other learning models such as k nearest neighbor (k-NN), support vector machine (SVM), and LSTM. Nonetheless, the proposed model obtained the best performance in terms of MAE, MAPE and RMSE, in all the prediction horizons.

Yipeng Liu et al. [7] have proposed a Conv-LSTM model for short-term traffic flow prediction. The main objective of using the Conv-LSTM model was to extract the spatial-temporal characteristics of the traffic flow. Furthermore, the authors also employed a bi-directional LSTM to obtain the traffic flow periodicity feature, by analyzing the historical data. The proposed method was evaluated using a real-world dataset and the results were compared with some existing learning models, namely ARIMA, Stack Denoise Autoencoder (SAE), LSTM, SVM, and CNN-LSTM. The dataset contained traffic data with a data collection frequency of 30 seconds and the experiment was done using data from two data detectors placed on a freeway and an urban road. MAE, RMSE and MAPE were used as the evaluation metrics and the proposed model achieved 4.408, 6.989 and 6.419 for the mentioned metrics respectively, the best performance compared to the existing methods.

Zhumei Wang et al. [8] proposed an approach for long-term traffic prediction using LSTM encoder-decoder architecture. The proposed architecture consists of two main parts, an attention-based encoder-decoder LSTM, and a calibration layer. The encoder-decoder LSTM enhances the neuronal memory and reduces the error propagation accumulation, while the calibration layer learns the group of the predicted data. The proposed model was trained and evaluated using the Caltrans performance measurement system dataset which is collected in Los Angeles with a data collection frequency of 5 minutes. Out of 1300 locations two were selected and divided into 60% training, 20% validation and 20% testing. Total RMSE and total symmetric MAPE were selected as the performance index. Five prediction horizons were experimented with, and the highest performance was produced by the proposed method with a total RMSE of 0.068 for 50 timesteps.

III. Methodology

The methodology of the proposed approach is discussed in this chapter. Section A discusses the dataset and the preprocessing techniques used before the learning process. Section B illustrates and discusses the proposed learning model architectures.

A. Data Description

The dataset used for the research was obtained from the open data portal of Austin, Texas. Optical traffic detectors that are deployed at signalized intersections and maintained by the arterial management division of the city of Austin transportation department, were used to collect data. The dataset contains traffic count and speed data collected using 56 optical traffic detectors. The data of the intersection with the most data points were selected for the analysis.

The data collection was carried out with a sampling rate of 15 minutes, hence, for each day there are 96 samples. Data are available from 1st of June 2017 to 14th April 2020, that is, 971 days, making the total 93216 samples. Fig. 2 plots the traffic volume for a week. As mentioned before a clear seasonality of 24 hours can be observed and the traffic behavior on weekends can be differentiated from the weekdays.

Deep learning models including LSTM networks, essentially make a mapping between the input features and the output prediction. There are different scales, depending on the units of the input features. Differences in the scales of the input features can make the forming of the aforementioned mapping difficult. Mostly this issue comes with features with large scales for which the models train larger weight values. Having large weights can result in unstable learning models that produce poor performances. To overcome this problem, one of the most common methods of preprocessing is a simple linear rescaling of input variables. Correspondingly, the output variables are post-processed to rescale to the original scale to obtain the required output [13].

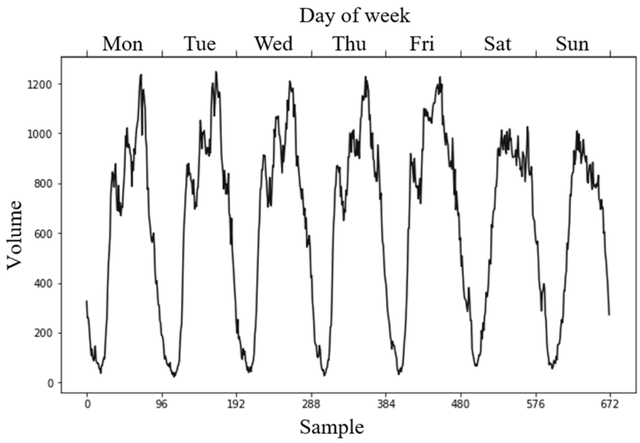

Fig 2. Traffic volume data of a week

In this research, we have used normalization where the variables are rescaled to the range 0 to 1, using the equation (7).

$$\acute{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7}$$

Here, the x and x' represents the original value and the normalized value respectively while x_{max} and x_{min} represent the maximum and minimum values of the series. Each variable was normalized and split into training and testing at the preprocessing stage. After the prediction, the predicted values were rescaled to the original scale using the same parameters used for the normalization of the output variable during the preprocessing.

The hour, day of the week and the past traffic volume values for the past 24 hours were selected as features. As mentioned before the seasonality of 24 hours shows the correlation between the hour and the traffic volume, as well as the variations during the weekends, convey that the traffic depends on the day of the week. This was also confirmed through a correlation analysis, where the hour and the day of the week showed a significantly high correlation. 60 % of the dataset was used for the training, leaving 40% for testing. 20% of the training data was used for the validation.

B. Learning Models

1) Standard LSTM model

The architecture used for the standard LSTM model is illustrated in Fig. 3. Three LSTM layers were used with 100, 50 and 100 hidden units, respectively. Finally, a dense layer consists of hidden units, the number of which is equal to the prediction horizon.

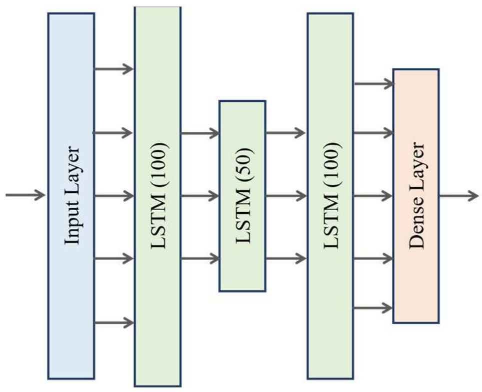

Fig 3. Standard LSTM model architecture

2) Encoder-decoder LSTM model

In the encoder-decoder model architecture, illustrated in Fig. 4., after the input layer, the LSTM layer of the encoder, followed by the repeat vector which then connects to the decoder LSTM layer. Both encoder and decoder layers have 100 hidden units. The output of the decoder layer goes through

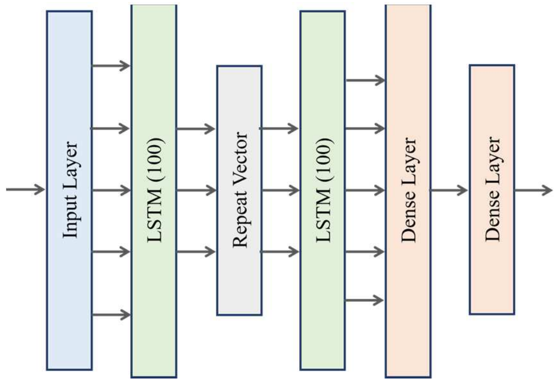

Fig 4. Encoder-decoder LSTM model architecture

a dense layer, before coming to the final dense layer that consists of hidden units, the number of which is equal to the prediction horizon. The dense layers that are used here are time distributed since it reduces the number of parameters or weights of the network since only one timestep is trained at a given time.

3) *CNN-LSTM model*

In the CNN-LSTM model, the encoder part of the encoder-decoder model is replaced with a CNN which consists of two convolutional layers with 64 filters and a max-pooling layer with a pool size of 2. Then before the repeat vector, the output of the pooling layer gets flattened. The decoder part of the model is the same as in the previous encoder-decoder model without any changes where the number of hidden units in the output dense layer is equal to the prediction horizon. The model architecture is illustrated in Fig .5.

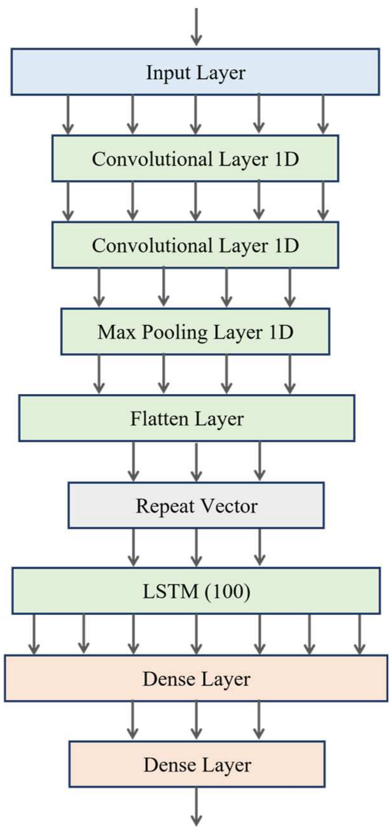

Fig 5. CNN-LSTM model architecture

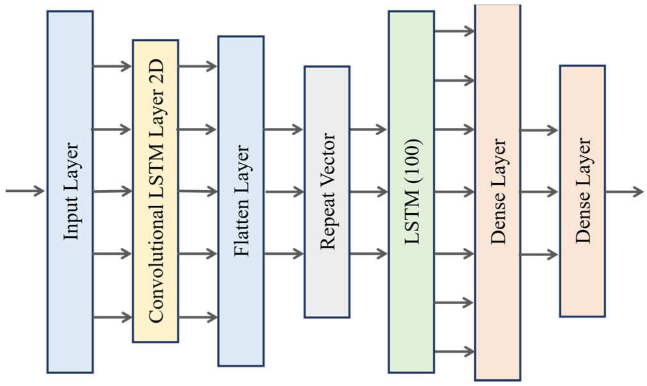

Fig 6. Conv-LSTM model architecture

4) *Conv-LSTM model*

The Conv-LSTM model architecture illustrated in Fig 6, has a 2D convolutional layer as the encoder which has 64 filters. Same as in the CNN-LSTM model, the output of the convolutional layer is flattened before the repeat vector.

C. *Evaluation*

Each learning model was trained and evaluated for eight prediction horizons, namely, 12, 6, 3, 2, and 1 hour, as well as 30 and 15 minutes. A callback function that monitors the validation loss, was used to prevent overfitting by stopping the training if no improvement in the validation loss was reported for five continuous epochs. Rolling window forecasting was used for the training as well as the evaluations, hence the data sequencing was performed accordingly.

Three error metrics namely, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) was used to evaluate the performance of the models. RMSE takes the residuals between actual and predicted values and compares prediction errors of different models for particular data. Giving high weights to large errors and scaling the scores in the same units as the forecasted values can be considered as the major characteristics of RMSE. Let A_i and F_i be the actual and the forecasted values respectively, for n observations, the RMSE is calculated using Equation (8).

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(A_i - F_i)^2}{n}} \tag{8}$$

Commonly, MAE is used for measuring an average model accuracy since it gives equal weights to the errors, calculated at each observation. Let A_i and F_i be the actual and the forecasted values respectively, for n observations, the MAE is calculated using Equation (9).

$$MAE = \frac{\sum_{i=1}^{n} |A_i - F_i|}{n} \tag{9}$$

MAPE is the average of absolute percentage errors which is popular in the industry since it is scale-independent and easy to interpret. Let A_i and F_i denote the actual and forecast values respectively for n observations, the MAPE is calculated using Equation (10).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{A_i - F_i}{A_i}\right| \times 100 \tag{10}$$

TABLE I. RESULTS OF THE EVALUATION

Prediction horizon	RMSE				MAE				MAPE (%)			
	Standard	***ED***	***CNN***	***Conv***	***Standard***	***ED***	***CNN***	***Conv***	***Standard***	***ED***	***CNN***	***Conv***
12hr (48)	96.36	96.64	83.15	85.19	67.75	68.09	57.34	59.57	17.26	17.44	14.71	15.30
6hr (24)	96.12	80.92	73.46	74.35	70.34	57.12	50.60	52.04	19.38	13.88	12.72	12.63
3hr (12)	73.81	80.87	67.64	67.97	52.07	58.28	46.81	47.47	13.17	15.18	11.15	11.81
2hr (8)	82.8	67.87	63.45	63.67	59.59	48.55	44.09	44.9	14.72	11.48	10.51	10.79
1hr (4)	59.09	59.39	58.03	64.98	42.38	43.10	41.14	46.69	10.96	10.61	9.72	10.70
30min (2)	55.29	59.53	53.28	65.14	40.49	44.34	38.2	46.98	10.51	12.07	9.26	11.77
15min (1)	50.09	51.08	50.87	49.23	36.58	37.63	36.99	36.07	9.31	9.62	9.29	9.03

D. Experimental Results

Table 1 presents the results obtained by each model, for a given prediction horizon. The terms Standard, ED, CNN, and Conv denotes standard LSTM, encoder-decoder LSTM, CNN-LSTM, and Conv-LSTM, respectively.

The prediction horizon column also denotes the number of samples that each prediction horizon includes, in the brackets. The results show that the performance of the models significantly increases when the prediction horizon decreases. Except for the 15-minute prediction horizon, the CNN-LSTM model has performed better than the other three. For the largest prediction horizon of 12 hours, the CNN-LSTM model has scored an RMSE of 89.15, MAE of 57.34, and a MAPE of 14.71. After the CNN-LSTM model, Conv-LSTM has the least errors compared to the standard LSTM and encoder-decoder LSTM and also has the best performance in 15-minute prediction horizon, scoring an RMSE of 49.23, MAE of 36.07, and a MAPE of 9.03%. The standard LSTM model has performed marginally better than the encoder-decoder LSTM model in terms of the errors.

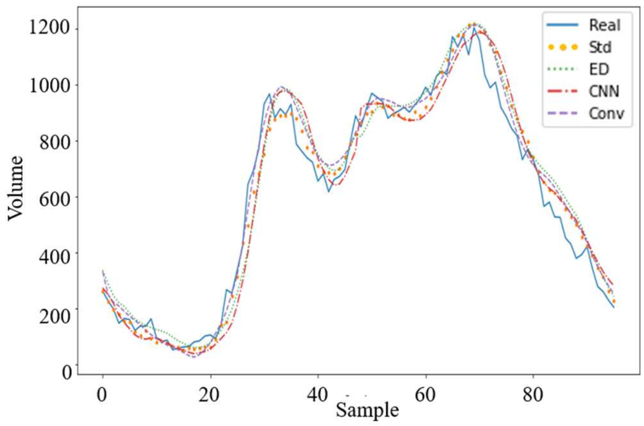

Fig 7. Prediction results for a weekday (Prediction horizon: 12 hours)

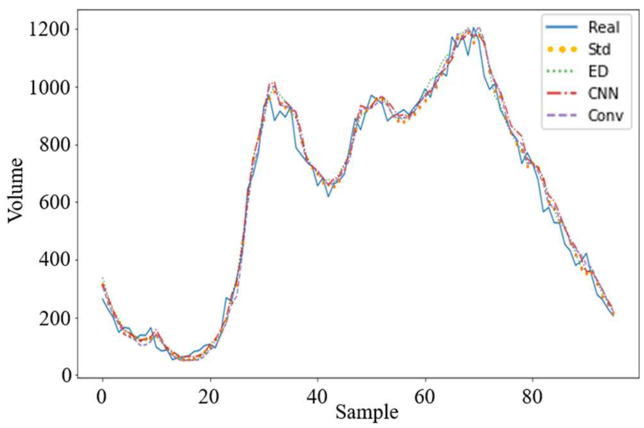

Fig 8. Prediction results for a weekday (Prediction horizon: 15 minutes)

IV. DISCUSSION

According to the experimental results, it is observed that the prediction horizon and the model performance are inversely proportional. We designed the models, to forecast values in the same sample frequency, hence when the prediction horizon is high, the number of samples that the model should predict is also high. For, example, in 12 hours prediction horizon, the model predicts 48 samples keeping a rate of 15 minutes, likewise, in 15 minutes prediction horizon the model only predicts one sample. Therefore, the decrease of the performance with the prediction horizon can occur as a result of the uncertainty increases with the prediction horizon.

CNN-LSTM and Conv-LSTM have produced better results, compared to the other two models. To further evidence the observations of the experimental results, the forecasted values of the four are plotted with the real traffic volume during a weekday, for the prediction horizons, 12 hours, and 15 minutes, in Fig. 7 and Fig. 8, respectively. In Fig. 7, only two model predictions have been used to cover the plot of the whole day since one prediction produces for 12 hours or 48 samples. It can be seen that the deviation of the 12 hours predictions is higher, compared to the predicted values for the 15-minute prediction horizon, plotted in Fig. 8. At the same time, Fig. 7 and Fig. 8 demonstrate that the CNN-LSTM and Conv-LSTM models produce better results, also resulting in smoother prediction curves compared to the other models.

Fig. 9 and Fig. 10 illustrate the prediction results during the Covid-19 pandemic. There was a clear anomaly in traffic behavior because of the travel restrictions and changes, caused by the pandemic, i.e., the traffic volume was decreased substantially. As mentioned before, the dataset in this research was divided into training and testing, where all the 2020 data were included in the testing set. Therefore, the models have never experienced this type of anomaly during the training. But we experimented and selected the models and the features to adapt to an anomaly such as this. And both Fig. 9 and Fig.10 proves that. Also, it is observed that the ability to adapt to an

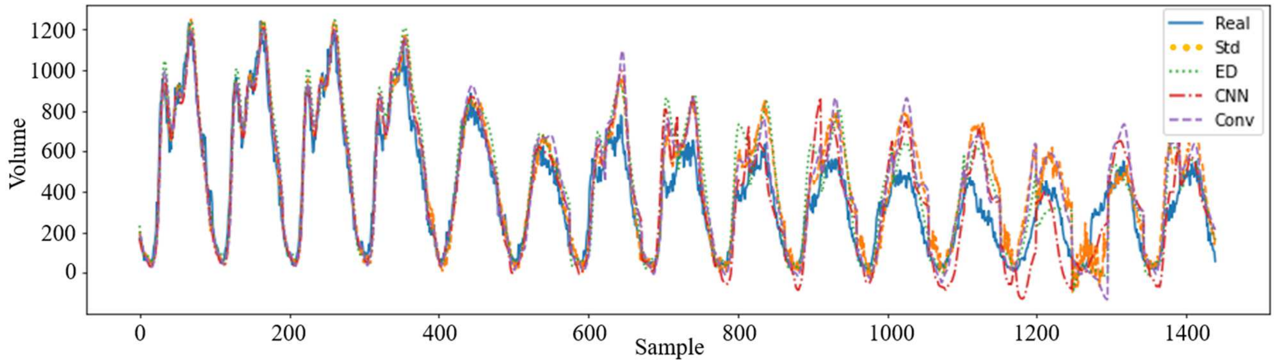

Fig 9. Prediction results during covid-19 period (Prediction horizon: 12 hours)

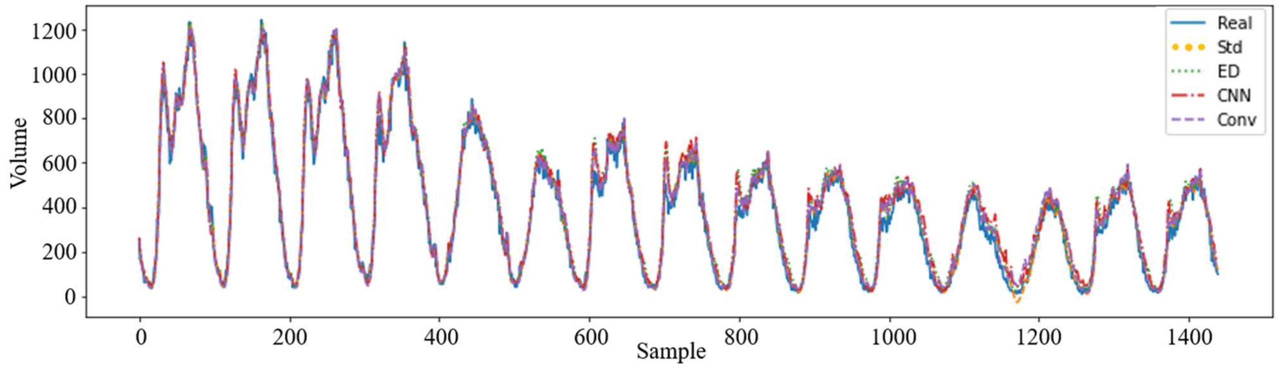

Fig 10. Prediction results during covid-19 period (Prediction horizon: 15 minutes)

anomaly also increases when the prediction horizon is reduced.

V. Conclusion

Short-term traffic volume forecasting has become a component with increasing usefulness in the traffic management domain. In this research, we have explored the possibility of using LSTM based models for short-term traffic forecasting. The models were evaluated for several prediction horizons and the Conv-LSTM model for the 15 minutes prediction horizon produced the best performance with an RMSE of 49.23 and a MAPE of 9.03%. Also, we observed the model behavior during the Covid-19 anomaly and the results have shown that the proposed LSTM models can be used efficiently in short-term traffic forecasting during an anomaly.

Acknowledgement

This research was supported by the Accelerating Higher Education Expansion and Development (AHEAD) Operation of the Ministry of Higher Education, Sri Lanka funded by the World Bank. Also, we acknowledge the support received from the Senate Research Committee Grant, University of Moratuwa, Sri Lanka.

References

[1] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4. pp. 1624–1639, Dec-2011.

[2] E. Vlahogianni, J. Golias, and M. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, pp. 533–557, 2004.

[3] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting traffic congestion using ARIMA modeling," in *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, 2019, pp. 1227–1232.

[4] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A Hybrid Deep Learning Model With Attention-Based Conv-LSTM Networks for Short-Term Traffic Flow Prediction," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–11, Jun. 2020.

[5] Z. Zhao, W. Chen, X. Wu, P. C. V. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Image Process.*, vol. 11, no. 1, pp. 68–75, Jan. 2017.

[6] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, "A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data," *Transp. Res. Part C Emerg. Technol.*, vol. 112, pp. 62–77, Mar. 2020.

[7] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with Conv-LSTM," in *2017 9th International Conference on Wireless Communications and Signal Processing, WCSP 2017 - Proceedings*, 2017, vol. 2017-January, pp. 1–6.

[8] Z. Wang, X. Su, and Z. Ding, "Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–11, Jun. 2020.

[9] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Futur. Comput. Informatics J.*, vol. 3, no. 2, pp. 334–340, Dec. 2018.

[10] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Mar. 2015.

[11] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 1724–1734.

[12] S. Du, T. Li, Y. Yang, X. Gong, and S. J. Horng, "An LSTM based Encoder-Decoder Model for MultiStep Traffic Flow Prediction," in *Proceedings of the International Joint Conference on Neural Networks*, 2019, vol. 2019-July.

[13] C. M. Bishop, *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc., 1995.