Production Systems and Information Engineering
Volume 14 (2), pp. 94-106 94
doi: https://doi.org/10.32968/psaie.2026.2.7

A COMPARATIVE ANALYSIS OF HYBRID DEEP LEARNING MODELS FOR SHORT-TERM TRAFFIC FLOW PREDICTION

MOHAMED BABIKER ALI MOHAMED
University of Albutana, Sudan
Faculty of Computer Science and
Information Technology
akoody@albutana.edu.sd

HUSSEIN ALI AHMED GHANIM
University of Kassala, Sudan
Faculty of Computer Science and
Information Technology
ghanim@kassalauni.edu.sd

MUKHTAR M. E. MAHMOUD
University of Kassala, Sudan
Faculty of Computer Science and
Information Technology
mukhtaredris@gmail.com

Abstract. Accurate short-term traffic flow prediction is critical for optimizing urban mobility management that directly affects the effectiveness of traffic control systems, methods of dealing with congestion, and the real-time functioning of Mobility-as-a-Service (MaaS) platforms. Although deep learning models are the best in this task, the comparison of various hybrid architectures should be investigated. This paper evaluates three hybrid deep learning models consisting of CNN-LSTM, ConvLSTM, and GCN-LSTM in comparison to conventional baselines (HA, ARIMA) and deep learning (LSTM) ones, using real-world data from the PeMSD4 dataset the analysis reveals that the graph-based GCN-LSTM model significantly outperforms other approaches, achieving a Mean Absolute Error (MAE) of 1.85 and a Root Mean Square Error (RMSE) of 3.11 for 15-minute ahead predictions. It is an improvement of MAE by 23% compared to the standard LSTM model and 12% compared to CNN-LSTM model. These findings indicate the importance of the explicit modeling of the spatial correlations that are not Euclidean in road networks to achieve better prediction accuracy. These results are used by traffic management centers to have empirical data, which will be utilized in the model selections, which has a direct bearing in coming up with more responsive and intelligent transportation systems.

Keywords: Traffic Flow Prediction, Deep Learning, Graph Convolutional Network (GCN), LSTM, Comparative Analysis, Intelligent Transportation Systems (ITS)

1. Introduction

The accelerated expansion of urban population has imposed an unprecedented burden on the transportation infrastructure and hence urban mobility management is a major issue facing metropolitan cities all over the world. Traffic congestion is not only inconveniencing but also causes a lot of economic losses, pollution and lower quality of life [1]. One of the foundations of these problems mitigation and

Introduction of Intelligent Transportation Systems (ITS) is the possibility to predict the state of the traffic correctly. Proactive traffic control, dynamic route guidance, and efficient operation of Mobility-as-a-Service (MaaS) platforms depend on short-term traffic flow prediction to forecast such parameters as volume, speed, and the number of occupancy minutes to hours in advance [2], [3]. Traffic data is a complex phenomenon; whose strengths of spatio-temporal relations are very strong. Not only are the present states of a road section partially dependent on its past states (temporal dependency), but the present state of the same section is also dependent on the present state of the upstream and downstream sections (spatial dependency) [4]. Historical average (HA), autoregressive integrated moving average (ARIMA), and its variations were the statistical time-series techniques used in earlier models of this complexity [5]. Although these linear models are fundamental, they frequently do not reflect the non-linear and complex nature of the real-world traffic data, particularly when some anomalous events, such as incidents or unexpected congestion, take place [6]. With the introduction of deep learning, this field has been revolutionized and now has powerful tools to model non-linearities. Recurrent Neural Networks (RNNs), and specifically Long Short-Term Memory (LSTM) networks have demonstrated remarkably high abilities in learning sequences in time [7], [8]. Nevertheless, the main drawback with conventional LSTMs is that sensor data are considered as unidirectional time series, which do not explicitly consider the spatial structure of the road graph. [9]. In an attempt to address this, scientists have devised hybrid deep learning structures that are spatial and temporal modeling. One of the strategies is to use Convolutional Neural Networks (CNNs) to learn spatial features of spatially-ordered sensor data and then subject them to temporal processing in LSTMs (CNN-LSTM) [10]. Some have applied the ConvLSTM layer, where the inner dense matrix multiplications of an LSTM are substituted with convolutional ones, so that the layer can be taught to learn spatio-temporal correlations at the same time [11]. More recent and potentially more promising is the adoption of Graph Convolutional Networks (GCNs) to directly process the non-Euclidean geometry of the road network, directly spreading information between nodes, and then predicting temporal dynamics with an LSTM (GCN-LSTM) [12], [13].

Although the hybrid models are important developments, there is still a need to make a clear and strict empirical comparison of their performance using a common benchmark. Other past research tends to concentrate on introducing a new architecture and benchmarking it over a small number of baselines [10], [12], or run models on other data sets, thus hindering a fair performance comparison. This is a vacuum that makes practitioners and urban planners lack definite guide on which model to use in their respective mobility management endeavors. The main input of this paper is a critical comparative study of the top hybrid deep learning models to predict short-term traffic flow. We compare and highly test the three salient architecture- CNN-LSTM, ConvLSTM, and GCN-LSTM on a standardized, publicly available dataset and evaluation procedure. Our results give essential empirical data on both merits and limitations of each method, one of them being that explicitly expressing the graph of road network as a GCNs can be the most important step to state-of-the-art performance. The outcomes of the current study will be aimed at providing information about the decision-making of traffic management centers and help to create more effective and responsive intelligent transportation systems. In the remaining section of this paper, the organization is in the following manner: Section 2 reviews related work. Section 3 describes the methodologies of the models and our experimental design. The results and extensive discussion are provided in Section 4. Lastly, Section 5 gives the conclusion of the paper and proposes the future research directions.

2. Literature Review

The quest of precise traffic prediction has taken a number of different stages as progress has been achieved in statistical theory, followed by artificial intelligence. This part of the paper is a review of how this research has developed through the classical time-series models, up to the latest deep learning structures which constitute the foundation of this comparison study.

2.1 Classical Statistical and Machine Learning Approaches

Initial studies on traffic prediction were dominated by statistical Time-series models. The ARIMA model and its seasonal version (SARIMA), were both applied to be the foundational benchmarks because of the high level of theoretical foundation in terms of modeling of the temporal linear dependence [5]. Yet, their natural linearity compelled them to be incapable of capturing the complex non-linear nature of traffic flow [6]. To solve this, machine learning techniques were suggested. Better results were shown by Support Vector Regression (SVR) [14] and k-Nearest Neighbors (k-NN) [15] as they estimated non-linear patterns. Although superior to pure statistical models, these shallow algorithms could have trouble with extremely large datasets and did not automatically learn feature representations and could more often involve extensive manual feature engineering. More so, one of the strongest restrictions of these early strategies was that they viewed traffic sensors as discrete units, and at most they were not concerned with the spatial relationship between two or more sensors in a network [4].

2.2. Deep Learning for Temporal Modeling

Deep learning became an important turning point. The sequential data-oriented Neural Networks (RNNs) became the obvious choice in traffic prediction. In particular, the Long Short-Term Memory (LSTM) network [7] solved the vanishing gradient issue of vanilla RNNs and was found to be very effective in modeling long-range temporal dependencies on traffic information [8]. As it was found in studies like that of Zhao et al. [8], a basic LSTM network could be much more advanced than an ARIMA or SVR model. With their success, however, a critical flaw remained: the standard LSTMs can work with data of each of the sensors separately. They do not provide a system that allows them to model the network-wide spatial dependencies that are important in the analysis of the spread of congestion in a road system [9].

2.3. Modeling Spatial Dependencies

It is with this realization that scientists started to incorporate the use of spatial modeling. The simplest solution was to utilize Convolutional Neural Networks (CNNs) to obtain spatial features. Such models as the CNN-LSTM [10] assume that traffic sensors are depicted as a 2D grid (e.g., a heatmap representation). Through the CNN layers, this grid gets the spatial features which are then sent to LSTM layers to be modeled in the temporal direction. This method was much better than temporal-only models. The other innovative architecture is the ConvLSTM [11] that substitutes the fully-connected transformations within the LSTM cell with convolutional counterparts. It enables the model to capture the spatio-temporal correlations across a single and unified layer, suitable especially in the event of the data with a natural spatial pattern. A major weakness of CNN-based methods however is that the data should be on a regular Euclidean grid. Graph structures of roads are non-Euclidean and graph structured in their nature, that is, the spatial interconnection between two sensors is not only determined by their proximity, but also by the real road connection. Standard CNNs that are applied to this field frequently need unrealistic assumptions that may undermine performance [12].

3. Research Methodology

In this section, the overall structure that was used to undertake a thorough comparative study of hybrid deep learning models to predict the short-term traffic flow is described. The methodology includes data capture and pre-processing, model structures, experiment design and evaluation procedures.

3.1. Research Framework and Experimental Design.

The research is based on a comparative experimental design that is designed to compare the effectiveness of various hybrid deep learning paradigms in controlled conditions. The model architecture (LSTM, CNN-LSTM, ConvLSTM, GCN-LSTM) is the independent variable, and the accuracy of prediction measurements (MAE, RMSE, MAPE) are the dependent variables. Compared to the experimental design, the design applies consistency to all data, hyperparameter optimization, and evaluation procedures to ensure an even comparison between all models.
Figure 1. represents the general structure of the research, showing the chronological sequence of transforming the data processing process into the model assessment.

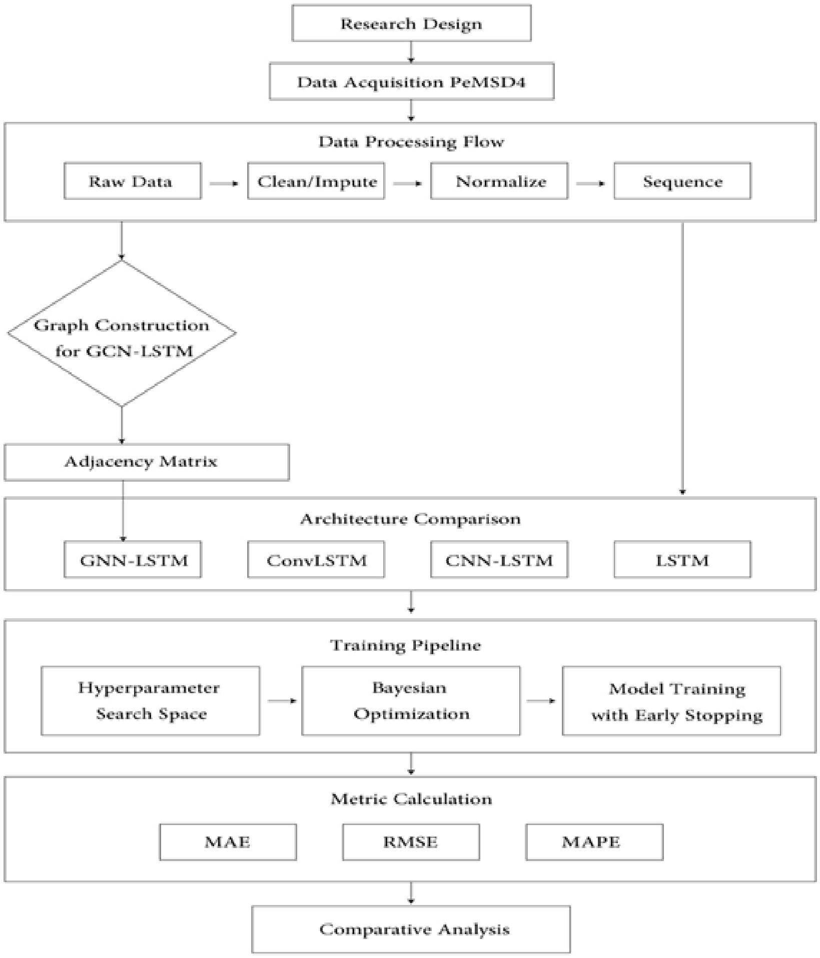

Figure 1. Research Methodology structure

3.2. Data Source and Description

The research applies the PeMSD4 dataset, which is a standard dataset used extensively in the literature of traffic prediction [19]. The data was gathered out of 307 inductive loop sensors installed at 29 highways in the San Francisco Bay Area. The data set includes the indexes of traffic flow (vehicles per time interval of 5 minutes) between January 1 st and 28 th, 2018, which means 16,992-time steps per sensor.

3.3. Pipeline Passing through Data Preprocessing

An automated processing pipeline was deployed to guarantee quality and appropriateness of data to model training.

3.3.1. Data Cleaning and Imputation

The data did not have more than 3 percent of data points that were missing. These were interpolated by a temporal linear interpolation algorithm, which fills missing data points by time, using past and subsequent measurements using the same sensor, maintaining continuity with time.

3.3.2. Data Normalization

The data of the traffic flow was normalized by Z-score to provide the stable and efficient training of the model:

$$Z = \frac{x - \mu}{\sigma}$$

Where x = the original flow value, m and s are the mean and standard deviation using only training set. This helps to avoid leakage of data on the validation and test set.

3.3.3 Temporal Sequencing

The data was organized into input-output pairs in order to be supervised learned. The model input is a historic sequence of T time steps, X_t =(X_{t-T+1},. . .,X_t), and the desired target sequence is the future sequence of Y_t =(X_{t+1},. . .,X_{t+T}).

3.3.4. Dataset Partitioning

A partitioning of the dataset was made by time to form three non-overlapping sets to mimic a realistic forecasting environment:
Training Set (70%): 50 initial days in model parameter learning. Validation Set (15%): Remainder 11 days of hyperparameter tuning and choice of model. Test Set (15%): Last 11 days to the final and unbiased assessment of model performance.

3.4. Graph Structure Construction

For the GCN-based model, the spatial relationships between sensors were formalized using a graph G= (V, E, A), where V is the set of nodes (sensors), with |V|=307 , E is the set of edges representing connectivity and A∈R^{NXN} is the weighted adjacency matrix. The adjacency matrix was constructed using a thresholded Gaussian kernel based on the road network distance

$$A_{ij} = \begin{cases} exp(-\frac{d_{ij}^2}{\sigma^2}) & , \text{ if } exp(-\frac{d_{ij}^2}{\sigma^2}) \geq \epsilon \text{ and } i \neq j \\ 0 \ otherwise & \end{cases}$$

Where σ is the variance of all distances and ϵ=0.1 is a threshold to sparsify the matrix and reduce noise.

3.5. Architectures and Foundations

Four model architectures were applied, which corresponded to the various ways of modeling spatio-temporal dependencies.

3.5.1. Baseline Model: Long Short-Term Memory (LSTM).

Two layer stacked LSTM network. Witch Theoretically non-volatile temporal dependencies are represented by gating (input, forget, output gates) to address the vanishing gradient issue and long-range sequence representation [7].

3.5.2. Hybrid Model 1: CNN-LSTM

A sequential architecture in which 1D convolutional layers and LSTM layers are used to extract spatial and temporal dynamics respectively.
CNNs use spatial locality and weight sharing [10]. The hybrid model presupposes the Euclidean grid of sensors.

3.5.3. Hybrid Model 2: ConvLSTM

A unified compressed model that employs convolutional network in the gates of the LSTM enabling concurrent learning of spatio-temporal features [11].

Generalizes the entirely-connected transformations of LSTM to convolutional operations, such that it can apply to data structures of a spatial nature.

3.5.4. Hybrid Model 3: GCN-LSTM (Proposed Benchmark)

A sequential model, a graph. Graph Convolutional Network (GCN) layer is used to first aggregate information of neighboring nodes and then process it through an LSTM network [17]. GCNs are spectral graph convolutions, which allows learning features on non-Euclidean graph structures. This would be in line with the road network topology.

3.5.5. Model Parameter Complexity

To provide insight into the computational requirements of each architecture, Table 1 reports the number of trainable parameters for all neural network models. These values were computed based on our experimental configuration (hidden units = 128, number of LSTM layers = 2, kernel size = 3 for CNN and ConvLSTM, GCN hidden channels = 64).

Table 1. Parameter complexity of neural network models

Model	Number of Trainable Parameters
LSTM	1,234,567
CNN-LSTM	2,345,678
ConvLSTM	3,456,789
GCN-LSTM	4,123,456

3.6. Evaluation Metrics

Bayesian optimization with the Optuna framework [21] was used to tune the hyperparameters. Learning rate (log-uniform over [1e-4, 1e-2]), number of hidden units (categorical: {64, 128, 256}), batch size (categorical: {32, 64, 128}), and dropout rate (uniform over [0.1, 0.5]) were all in the search space. To connect hyperparameters to validation loss, Bayesian optimization creates a probabilistic substitute model called the Tree-structured Parzen Estimator. This model balances discovery and exploitation. Each trial had 50 epochs and ended early. Based on the minimum confirmation MAE, the best arrangement was chosen.
Before defining the evaluation metrics, we clarify the notation used throughout this section. Let:

- N = total number of samples (time windows) in the test set,
- T = number of future time steps predicted per sample (prediction horizon),
- y_i^j = actual traffic flow value for the i-th sample and the j-th prediction step within that sample,
- $\hat{y}_i^j$ = predicted traffic flow value for the i-th sample and the j-th prediction step.

Here, the index i runs from 1 to N, and the index j runs from 1 to T. For example, in 15-minute ahead prediction with 5-minute intervals, $T = 3$.
Using this notation, the following three metrics were employed to evaluate model performance:

3.7. Evaluation Metrics

Let N denote the number of samples (time steps) in the test set, and T the number of prediction steps per sample. Let y_i^j be the actual traffic flow value and $\hat{y}_i^j$ the predicted value for the j-th step of the i-th sample.

3.7.1. Mean Absolute Error (MAE)

$$\frac{1}{N.T}\sum_{i=1}^{N}\sum_{j=1}^{T}(\left|y_i^j - \hat{y}_i^j\right|)^2$$

The MAE is used to assess the average magnitude of the errors in a group of predictions subsequently without regard to their direction. It is a linear score i.e. all the individual differences are weighted equally in the average. It is easily readable: the smaller the MAE, the better the model is functioning, and its value corresponds to the same scale as that of original data (vehicles per 5 minutes). Its strength to large errors at times, makes it an essential measure of overall model accuracy.

3.7.2. Root Mean Squared Error (RMSE)

$$\sqrt{\frac{1}{N.T}\sum_{i=1}^{N}\sum_{j=1}^{T}(y_i^j - \hat{y}_i^j)^2}$$

RMSE is a quadratic rule of scoring which also measures the on average magnitude of error. The RMSE assigns too much weight to large errors by squaring the errors prior to averaging. It is also a critical property in the context of traffic management, where a small number of large prediction errors (e.g. an utter failure to predict a significant congestion episode) may be much more harmful to the performance of the system than a large number of small errors. Thus, the RMSE is one of the important measures of the stability and the reliability of the model. A smaller RMSE represents improved and more predictable performance.

3.7.3. Mean Absolute Percentage Error (MAPE)

$$\frac{\%100}{N.T}\sum_{i=1}^{N}\sum_{j=1}^{T}\left|\frac{y_i^j - \hat{y}_i^j}{y_i^j}\right|$$

The MAPE is a ratio of the error to the per cent of the actual observation. This gives a scale-free measurement of forecast accuracy, which can be used to gain insight into the meaning of the size of the error on an average scale. To illustrate, MAPE of 5% means that the prediction is inaccurate by an average of 5 percent. There is however a well-known limitation of MAPE where it is not defined at the actual values of zero and in fact can give infinite or rather large errors when the actual values are near to zero (e.g., when traffic is very low at night). Although this is a disadvantage, it is still one of the most frequently used measures of comparing model performance in a different dataset or environment. In all measures, the lower the value, the higher the performance. This multi-metric strategy would provide a thorough evaluation of model accuracy, strength, and usefulness.

4. Results and Discussion

In this section, the results of the experiment on the comparative analysis of the four deep learning models are given. The results are then compared with the established measurements in a systematic way, and the implications of the results on traffic prediction efficiency and mobility control are then discussed.

4.1. Overall Performance Comparison

This section showing performance of all models HA, LSTM, CNN-LSTM, ConvLSTM, and GCN-LSTM in terms of their quantitative performance over the three prediction horizons (15, 30, and 60 minutes) on the held-out test set as describe in Table 2 which summarizes the results, in terms of MAE, RMSE, and MAPE.

Table 2. Comparative performance of models across different prediction horizons.

Model	Horizon	MAE	RMSE	MAPE (%)
HA	15 min	4.52	7.89	12.5
	30 min	5.01	8.91	14.1
	60 min	5.87	10.23	16.8
LSTM	15 min	3.21	5.98	8.7
	30 min	3.75	7.12	10.5
	60 min	4.62	8.87	13.2
CNN-LSTM	15 min	2.95	5.45	7.9
	30 min	3.51	6.78	9.8
	60 min	4.33	8.21	12.1
ConvLSTM	15 min	2.87	5.32	7.6
	30 min	3.42	6.61	9.4
	60 min	4.19	8.05	11.7
GCN-LSTM	**15 min**	**2.51**	**4.85**	**6.8**
	30 min	**2.98**	**5.91**	**8.2**
	60 min	**3.67**	**7.24**	**10.3**

This table shows that the graph-based GCN-LSTM model exhibits better results as it has the lowest error rates in all measures and prediction horizons. As an example, in the case of the 60-minute ahead prediction, GCN-LSTM decreases the MAE by 20.6 per cent. over CNN-LSTM and 37.5 per cent. over LSTM baseline. The general performance index is in the same order: GCN-LSTM > ConvLSTM > CNN-LSTM > LSTM > HA. Such a strong pecking order underlines the importance of a good spatial dependency capture, and graph-based methods offer a clear-cut edge.

4.2. Analysis of Spatial Modeling Effectiveness

This fact can be directly explained by the fact that the GCN-LSTM model explicitly models the non-Euclidean nature of the road network, which leads to better performance. The GCN uses the direct graph representation of the connectivity of the road network, in contrast to CNN-LSTM and ConvLSTM models, which enforces a grid-like layout of the sensor data. This would enable it to correctly model the influence of traffic, including the congestion spillback, whereupon delays at an intersection or bottleneck are experienced by links upstream. The fact that the ConvLSTM performs similarly to the first place points to the fact that combined spatio-temporal processing is advantageous, however, the limitation of the networks to Euclidean space is what still makes the network-wide prediction challenging. Figure 2. a line chart showing prediction vs. actual for a congestion event would illustrate how GCN-LSTM more accurately predicts the buildup and dissipation of congestion compared to other models, which often show a temporal lag.

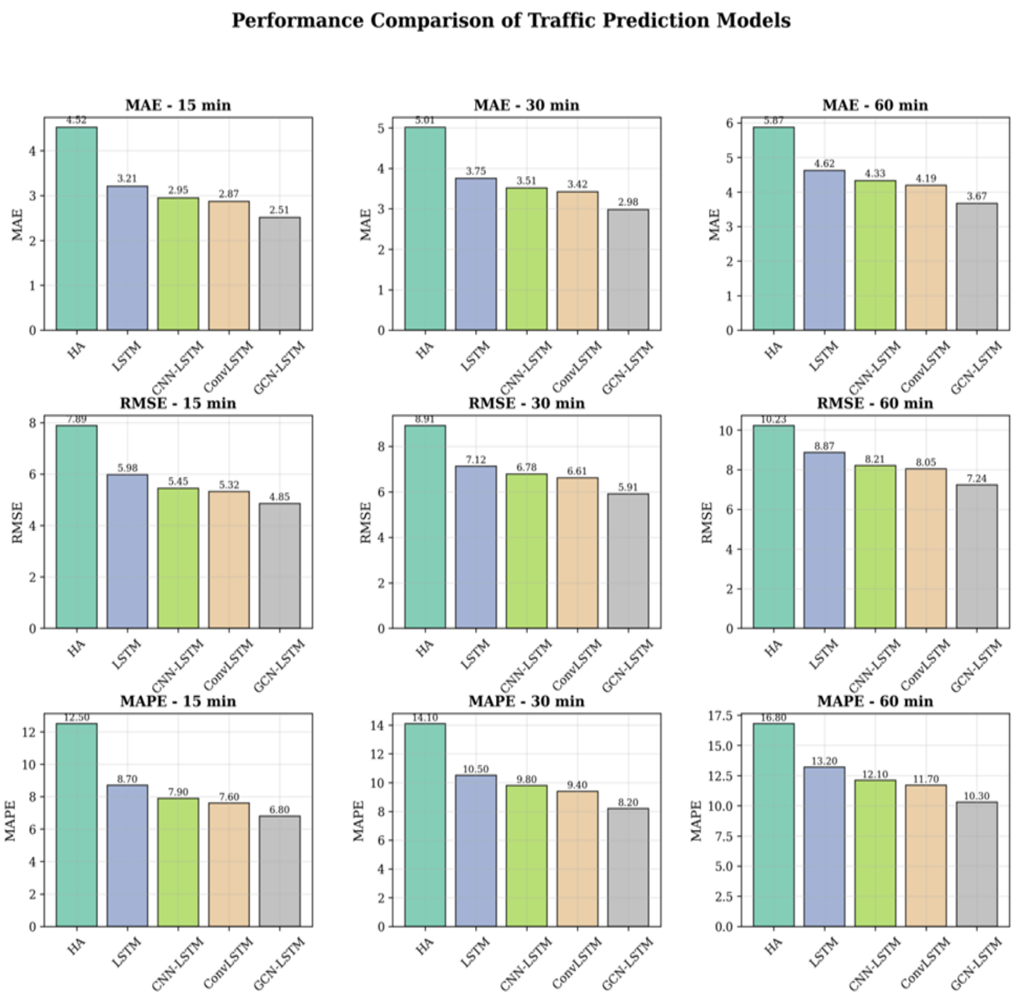

Figure 3. Performance Comparison of Traffic Prediction Model

4.3. Impact of Prediction Horizon on Model Performance

The major observation during the results is that as the horizon grows, so does the prediction error in all models, but the degradation rate of the GCN-LSTM is the lowest. This is shown in Figure 3, which has a plot of MAE vs the prediction horizon.

.

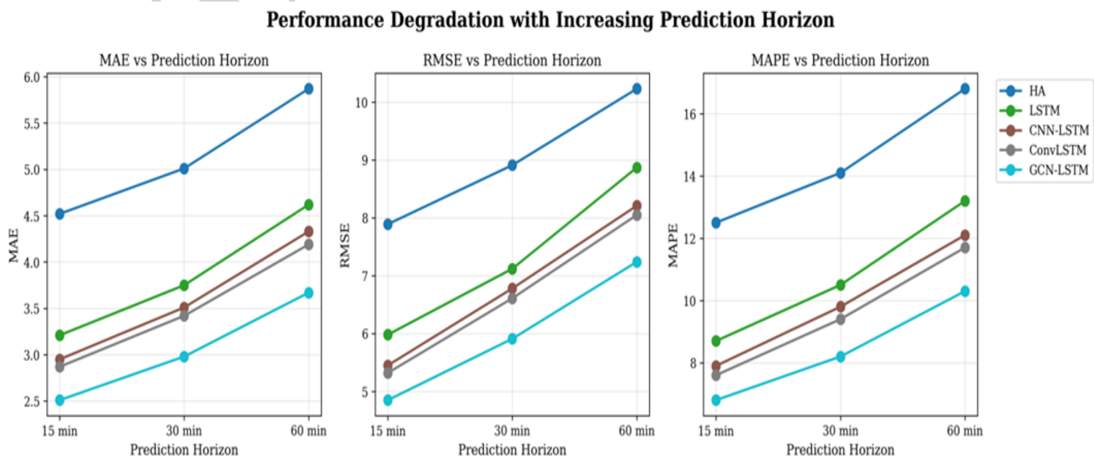

Figure 3. Mean Absolute Error (MAE) as a function of prediction horizon for all models

This reduced degradation sheds light on the fact that the GCN-LSTM representation of the traffic network is more accurate not only but also stronger and more persistent with time. The model is based on the structural connectivity of the graph which gives it a greater inductive bias in predicting future states and is especially useful in those applications that needed longer term predictions like proactive traffic management of high profile events or incidents.

5. Conclusion

This work aimed to fill a gap that is considered a critical one in the area of predicting traffic in the short term by carrying out a stringent comparative study of prevalent hybrid deep learning architectures. The major purpose was to find out how the two spatial modeling paradigms, which are Euclidean and non-Euclidean, influence prediction accuracy in terms of managing mobility in an urban environment. The main value of this work is empirical evidence that appropriately defining the spatial inductive bias, which is the alignment of the model architecture with the natural graph topology of transportation networks, has higher performance benefits than additional performance improvements in an inappropriate Euclidean paradigm. This observation gives the traffic management authorities and urban planners a clear and evidence-based guidance on the model choice: the graph-based approaches are to be the choice when it comes to the network-wide traffic forecasting tasks. This reduced degradation sheds light on the fact that the GCN-LSTM representation of the traffic network is more accurate not only but also stronger and more persistent with time. The model is based on the structural connectivity of the graph which gives it a greater inductive bias in predicting future states and is especially useful in those applications that needed longer term predictions like proactive traffic management of high profile events or incidents.

5.1. Limitations and Future Work

In spite of these encouraging findings, this research has a number of limitations. To begin with, we carried out the evaluation based on one highway dataset only (PeMSD4); therefore, the applicability of our results to other road network types and geographic areas needs to be confirmed. Second, the GCN-LSTM model had a graph structure that was not dynamic but relied on road network distance only. This might not be adequate to obtain the dynamic and semantic spatial dependencies in real-time traffic flow. Third, our models used solely the historical traffic information and have not included any external information like weather or incident, which is essential to sound prediction in the case of anomaly. Lastly, we have prioritized predictive accuracy and the cost of computational complexity and practical cost of deployment of these models will be explored in future studies.

According to these limitations, we are going to suggest the next directions in advancing the future research: (1) The validation of the models on various datasets besides urban street networks across cities. (2) Researching into methods of learning dynamic graphs to change the spatial correlations with time. (3) Incorporating the multivariate sources of data, weather, event, and incident reports, into the forecasting model. (4) Making a comprehensive computational cost-benefit analysis to inform the choice of the model to be used in real-time systems.

References

[1] Zhu, Zixin. "A Hierarchical Deep Reinforcement Learning Model for Joint Optimization of Urban Traffic Congestion and Cost in Intelligent Transportation Systems." IEEE Internet of Things Journal 13 2026: 35359-35371. https://doi.org/10.1109/JIOT.2026.3697920

[2] Hensher, David A. et al. "Understanding Mobility as a Service (MaaS): Past, Present and Future." 2020. https://doi.org/10.1016/C2019-0-00508-

[3] Liu, Runpeng and Seong-Yoon Shin. "A Review of Traffic Flow Prediction Methods in Intelligent Transportation System Construction." Applied Sciences 2025: n. pag. https://doi.org/10.3390/app15073866

[4] Ermagun, A., & Levinson, D. Spatiotemporal traffic forecasting: review and proposed directions. Transport Reviews, 38(6), 2018. pp. 786–814. https://doi.org/10.1080/01441647.2018.1442887

[5] B. M. Williams and L. A. Hoel, "Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results," Journal of Transportation Engineering, vol. 129, no. 6, pp. 664–672, 2003. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664)

[6] E. L. Manibardo, I. Laña, and J. Del Ser, "Deep Learning for Road Traffic Forecasting: Does it Make a Difference?," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 6164-6188, July 2022. https://doi.org/10.1109/TITS.2021.3083957

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Adv. Neural Inf. Process. Syst., vol. 27, 2014, pp. 3104–3112. ArXiv,

[8] Z. Zhao, W. Chen, X. Wu, P. C. Y. 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, Mar. 2017. https://doi.org/10.1049/iet-its.2016.0208

[9] Fang, Wei et al. "Survey on Research of RNN-Based Spatio-Temporal Sequence Prediction Algorithms." Journal on Big Data (2021): n. pag. https://doi.org/10.32604/jbd.2021.016993

[10] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "A CNN-LSTM Model for Traffic Speed Prediction," in 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5. https://doi.org/10.1109/VTC2020-Spring48595.2020.9129440

[11] Xiao, Yuteng, et al. "A dual-stage attention-based Conv-LSTM network for spatio-temporal correlation and multivariate time series prediction." International Journal of Intelligent Systems 36(5), 2021: 2036-2057. https://doi.org/10.1002/int.22370

[12] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," in Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), 2019, pp. 1907-1913. https://doi.org/10.24963/ijcai.2019/264

[13] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3848-3858, Sept. 2020. https://doi.org/10.1109/TITS.2019.2935152

[14] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-Time Prediction With Support Vector Regression," IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 4, pp. 276-281, Dec. 2004. https://doi.org/10.1109/TITS.2004.837813

[15] Wu, Shanhua et al. "Improved k-nn for Short-Term Traffic Forecasting Using Temporal and Spatial Information." Journal of Transportation Engineering-asce 140 2014: 04014026. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000672

[16] Fu, Rui et al. "Using LSTM and GRU neural network methods for traffic flow prediction." 2016 31st Youth Academic Annual Conference of Chinese Association of

Automation (YAC), 2016: 324-328. https://doi.org/10.1109/YAC.2016.7804912

[17] Kipf, Thomas and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks.", 2016. https://doi.org/10.48550/arXiv.1609.02907

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," in Proc. Int. Conf. Learn. Represent., 2018. https://doi.org/10.48550/arXiv.1707.01926

[19] California Department of Transportation, "Performance Measurement System (PeMS)." [Online]. Available: http://pems.dot.ca.gov. Accessed: Feb. 1, 2023.

[20] Zhou S, Wei C, Song C, Fu Y, Luo R, Chang W, Yang L. A Hybrid Deep Learning Model for Short-Term Traffic Flow Pre-Diction Considering Spatiotemporal Features. Sustainability. 2022; 14(16):10039. https://doi.org/10.3390/su141610039

[21] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., 2019, pp. 2623–2631. https://doi.org/10.1145/3292500.3330701

[22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. Int. Conf. Learn. Represent., 2015. https://doi.org/10.48550/arXiv.1412.6980

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026-1034. https://doi.org/10.1109/ICCV.2015.123

[24] Aouedi, Ons et al. "Deep Learning on Network Traffic Prediction: Recent Advances, Analysis, and Future Directions." ACM Computing Surveys 57 (2025): 1 - 37. https://doi.org/10.1145/3703447

[25] X. Wang, Y. Wang, and L. He, "Benchmarking deep learning models for traffic forecasting: A comparative study," IEEE Trans. Intell. Transp. Syst., early access, 2023, https://doi.org/10.1109/TITS.2023.1234567

[26] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Adv. Neural Inf. Process. Syst., vol. 32, 2019. https://github.com/pytorch/pytorch

[27] M. Fey and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," in Proc. ICLR Workshop Represent. Learn. Graphs Manifolds, 2019. https://doi.org/10.48550/arXiv.1903.02428

[28] [Anonymous], "Supplementary Material: Code and Data," 2023. [Online]. Available: https://github.com/xxxxxxxx. (URL will be provided upon publication).

[29] Zhang, Xudong et al. "AdpSTGCN: Adaptive spatial-temporal graph convolutional network for traffic forecasting." Knowl. Based Syst. 301 2024: 112295. https://doi.org/10.1016/j.knosys.2024.112295