

GNN-Based Traffic Flow Forecasting and Its Future Scope in Nepal

¹Dikshya Aryal, ^{2*}Hemant Joshi

^{1,2}*Department of Computer Engineering, Universal Engineering and Science College, Lalitpur, Nepal*

Email: dikshyauesc@gmail.com

**Corresponding author: hemantajoshi@uesc.edu.np*

Abstract

In modern urban environments, traffic congestion poses a significant challenge for intelligent transportation systems, which demand accurate and scalable traffic flow forecasting solutions. Conventional time series approaches fail to capture the spatial dependencies inherent in road networks, which motivates the use of graph-structured models. This paper presents a comparative evaluation of three approaches: Linear Regression, Long Short-Term Memory (LSTM) networks, and Spatio-Temporal Graph Convolutional Network (STGCN) for short-term traffic speed forecasting on the METR-LA benchmark dataset, which consists of 207 highway sensors. The models are evaluated using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) at a 15-minute prediction horizon. As a result, STGCN achieves an MAE of 2.18 mph and RMSE of 3.75 mph, outperforming the LSTM baseline by 29.7 % on MAE and Linear Regression by 18.4% on MAE, showing that jointly modeling spatial road network topology and temporal traffic dynamics significantly improves forecasting performance. Finally, we analyze the applicability of graph-based traffic forecasting to the rapidly urbanizing road networks of Nepal, where the lack of sensor infrastructure and data scarcity present open research challenges for future intelligent transportation systems.

Keywords: *Traffic flow forecasting, Graph Neural Network, STGCN, METR-LA, Intelligent Transportation Systems.*

I. INTRODUCTION

Traffic congestion has become one of the most pressing infrastructure challenges of the modern era over the past two decades due to the rapid urbanization. The economic costs of urban congestion are substantial, commuters in large metropolitan areas lose significant productive hours annually to traffic delays [1]. In cities such as Kathmandu, the problem is acutely compounded by unplanned road infrastructure, narrow arterial roads, and a vehicle fleet that has grown significantly faster than road capacity. Intelligent Transportation Systems (ITS) is a fundamental enabling technology for an accurate short-term traffic flow forecasting, predicting traffic volume 15 to 60 minutes ahead. Reliable forecasts power adaptive traffic signal control, real-time route guidance, incident detection, and fleet management systems

[2]. The quality of these downstream applications is directly tied to forecasting accuracy, making model selection a decision of practical engineering consequences. The traffic forecasting literature has evolved through three broad generations. Statistical time-series models, particularly ARIMA (Autoregressive Integrated Moving Average) and its variants [3], dominated early research but are limited to linear dynamics and cannot represent spatial interactions between road segments. The second generation introduced recurrent architectures, mostly notably LSTM networks [4], which capture complex nonlinear temporal dependencies but treat each road sensor as an independent time series, entirely ignoring road network topology. The third generation addresses this limitation through Graph Neural Network (GNN) [5], which represent road networks as graphs where sensors are nodes and road connections are edges, enabling spatial dependency modeling alongside temporal pattern learning.

This paper contributes:

1. To systematic quantitative performance of Linear Regression, LSTM, and STGCN on the METR-LA benchmark,
2. To a detailed methodological exposition of each model, including graph construction and the message-passing mechanism,
3. To a structured analysis of the future research scope of GNN-based forecasting in Nepal.

II. LITERATURE REVIEW

Early traffic forecasting research relied on statistical time-series models. Ahmed and Cook [3] demonstrated the applicability of Box-Jenkins ARIMA models to freeway traffic flow. Williams and Hoel [6] showed that seasonal ARIMA models better capture the periodicity of daily and weekly traffic. Linear Regression has been widely used as a feature-based baseline in several comparative studies [7], capturing aggregate relationships between historical observations and future flow without modeling sequential dependencies. While computationally efficient, these methods are fundamentally limited by their linearity assumptions and inability to model cross-sensor spatial interactions.

Hochreiter and Schmidhuber [4] introduced LSTM networks, whose gating mechanism: the input gate, forget gate, and output gate address the vanishing gradient problem in standard recurrent networks. Ma et al. [8] applied LSTM to short-term traffic speed prediction, demonstrating superior accuracy over ARIMA baselines. Despite strong temporal modeling capability, LSTM treats each road sensor independently, unable to propagate congestion information spatially across the network.

Kipf and Welling [5] introduced Graph Convolutional Network (GCN), generalizing convolution to graph-structured data. Li et al. [9] proposed DCRNN (Diffusion Convolutional Recurrent Neural Network), modeling traffic dynamics as a diffusion

process on the directed road graph combined with an encoder-decoder LSTM for multi-step forecasting. Yu et al. [10] proposed STGCN, replacing recurrent layers with gated temporal convolutions, achieving faster training. Wi et al. [11] developed Graph WaveNet with a self-learning adaptive adjacency matrix. Zhao et al. [12] proposed T-GCN (Temporal Graph Convolutional Network) combining GCN with Gated Recurrent Unit (GRU) for lightweight spatial-temporal modeling. Our work uses STGCN as the representative graph-based model and provides a systematic comparison against classical and deep learning baselines on the METR-LA benchmark.

III. METHODOLOGY

A. Dataset: METR-LA

The METR-LA dataset [9] contains readings from 207 loop detector sensors on the Los Angeles metropolitan highway network, which were collected at 5-minute intervals between March 2012 and June 2012, generating 34,272 time steps per sensor. Traffic speed is reported in miles per hour (mph). We follow the standard 70/10/20 train/validation/test split. Input sequences consist of $T = 12$ historical time steps (60 minutes of history), and the target is the speed at the next time step (15 minutes ahead prediction). Missing sensor values are filled using linear interpolation, and all speed values are normalized to zero mean and unit variance using statistics computed solely from the training split to prevent data leakage.

Table 1: METR-LA Dataset Statistics

Property	Value
Number of sensors	207
Sampling interval	5 minutes
Total time steps per sensor	34,272
Input history length (T)	12 steps (60 minutes)
Prediction horizon	1 step (15 minutes)
Train/Val/Test split	70% / 10% / 20%

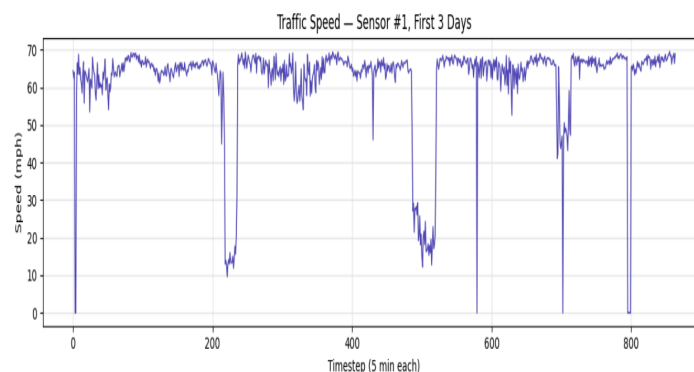


Fig. 1: Traffic speed signal of a representative sensor over 72 hours

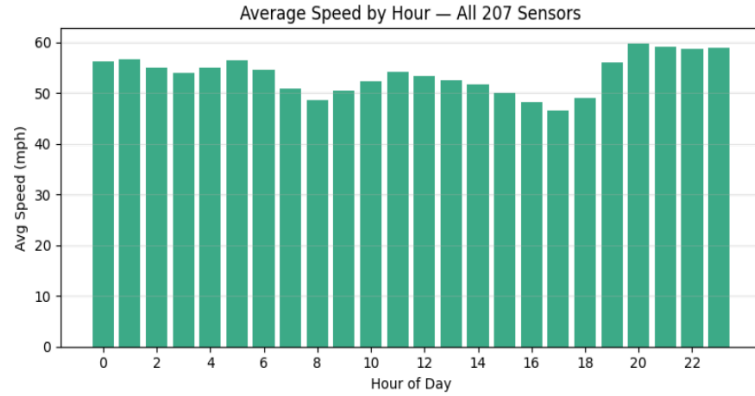


Fig. 2: Hourly Average Speed of the day across all 207 sensors in the METR-LA dataset

B. Problem Formulation

Let the road network be represented as a weighted graph $G = (V, E, A)$, where V is the set of $N = 207$ sensor nodes, E is the set of road edges, and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix encoding road connectivity. At each time step t , each node v_i has a scalar traffic speed observation $x_i^t \in \mathbb{R}$. Given G and T are historical observations $X = \{X^{(t-T+1)}, \dots, X^{(t)}\}$, the forecasting objective is: $f(G; X^{(t-T+1)}, \dots, X^{(t)}) \rightarrow X^{(t+1)}$. The adjacency matrix is constructed using a thresholded Gaussian kernel on pairwise road network distances [9]: $A_{ij} = \exp(-d_{ij}^2/\sigma^2)$ if $d_{ij} < \kappa$, else 0, where d_{ij} is the road distance between sensors i and j , σ^2 is the variance of all pairwise distances, and κ is a sparsity threshold. The adjacency matrix shows a sparse connectivity pattern which reflects the localized interactions among neighboring traffic sensors. Fig.3 shows the adjacency matrix heatmap for the first 30 sensors, where brighter values indicate stronger road connectivity.

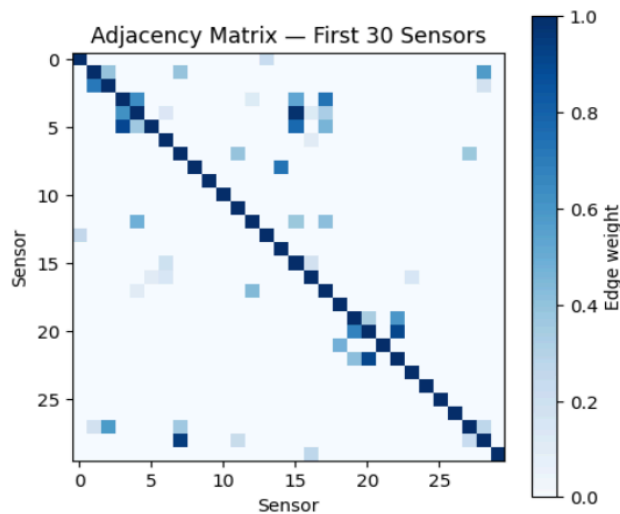


Fig. 3: Adjacency matrix heatmap of the first 30 sensors in the METR-LA road network graph

C. Model 1: Linear Regression

Linear Regression serves as the classical baseline, and for each independent sensor, a linear model is trained to predict the next-time step speed using $T = 12$ historical values as input features: $\hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_{12} x_{12}$, where x_i denotes the traffic speed observed at the i -th historical time step (equivalently, at time $t - T + i$) and the coefficients $\beta_0, \beta_1, \dots, \beta_{12}$ are estimated by Ordinary Least Squares (OLS). Despite its simplicity, Linear Regression effectively exploits the strong periodic patterns in highway traffic data, in which the same time slot from the previous day is a strong predictor of current conditions. This makes it a competitive baseline on structured highway datasets such as METR-LA. In this paper, Linear Regression is included as a traditional statistical baseline to quantify the improvements obtained from increasingly sophisticated temporal and graph-based learning architectures.

D. Model 2: Long Short-Term Memory (LSTM)

LSTM networks [4] use gating mechanisms to selectively retain or discard information across time steps, by addressing the vanishing gradient problem in standard recurrent networks. At each time step, the LSTM computes the following operations, where $\sigma(\cdot)$ denotes the sigmoid activation function and $\odot$ denotes element-wise multiplication throughout:

Forget gate: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$

Input gate: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

Candidate cell state: $\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$

Cell update: $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$

Hidden state: $h_t = o_t \odot \tanh(C_t)$.

In this study, a two-layered stacked LSTM with a hidden dimension 128, batch normalization, and dropout ($p = 0.3$) is applied to each sensor's 12-step input sequence. The model is trained using the Adam optimizer with learning rate of 0.0003 and Huber loss for robustness to speed outliers. While LSTM captures temporal dynamics effectively, it has no mechanism for inter-sensor communication. Congestion propagating from one node to adjacent nodes cannot be represented.

E. Model 3: STGCN

STGCN [10] is a purely convolutional spatial-temporal architecture that avoids recurrent layers, using graph convolution for spatial encoding and gated 1D convolution for temporal encoding. The core spatial module applies Chebyshev graph convolution:

$$g * x \approx \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L}) x$$

where, T_k is the k -th Chebyshev polynomial on the scaled Laplacian $\tilde{L} = 2L/\lambda_{\max} - I$, and θ_k are learnable filter coefficients. The temporal module uses gated convolution:

$$\Gamma * Y = P \odot \sigma(Q), \text{ where } [P, Q] = \Gamma * Y.$$

The full architecture consists of two ST-Conv blocks; each is a sandwich of temporal convolution, graph convolution, and temporal convolution, followed by batch

normalization, ReLU, and a linear output layer. With only 10,609 parameters, STGCN is significantly more compact than LSTM (331,599 parameters) while achieving superior accuracy by using road network topology.

F. Evaluation Metrics

In this paper, model performance is assessed using two standard regression metrics. MAE measures average absolute prediction error:

$$\text{MAE} = (1/n) \sum_i |y_i - \hat{y}_i|$$

RMSE penalized large errors more heavily than MAE:

$$\text{RMSE} = \sqrt{(1/n) \sum_i (y_i - \hat{y}_i)^2}$$

RMSE gets larger errors than MAE, which makes it sensitive to peak-hour forecasting failures. Both metrics are computed on denormalized predictions in the original mph space where lower values indicate better forecasting performance for both MAE and RMSE.

IV. EXPERIMENTS AND RESULTS

A. Implementation Details

All models are implemented in Python using the PyTorch framework. STGCN uses PyTorch Geometric for graph convolution operations. Experiments are conducted on a GPU-accelerated environment (Google Colab, NVIDIA T4). Therefore, Table 2 summarizes the hyperparameter configuration.

Table 2: Hyperparameter Configuration for All Models

Hyperparameter	Linear Reg.	LSTM	STGCN
Optimizer	OLS	Adam	Adam
Learning Rate	N/A	0.0003	0.001
Batch size	N/A	32	32
Hidden dim	N/A	128	64
Parameters	~2,500	331,599	10,609
Loss function	MSE	Huber loss	MSE
Max epochs	N/A	150	100
Early stopping	N/A	20 epochs	15 epochs

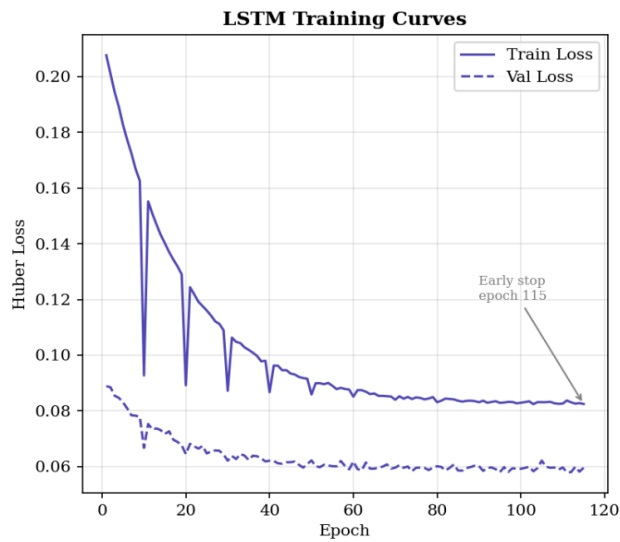


Fig. 4: Training and validation loss curve for the LSTM model over 115 epochs

B. Quantitative Results

Table 3 presents the forecasting performance of all three models on the METR-LA test set at the 15-minute prediction horizon.

Table 3: Model Performance on METR-LA Test Set

Model	MAE (mph)	RMSE (mph)
Linear Regression	2.67	4.07
LSTM	3.10	5.27
STGCN (Proposed)	2.18	3.75

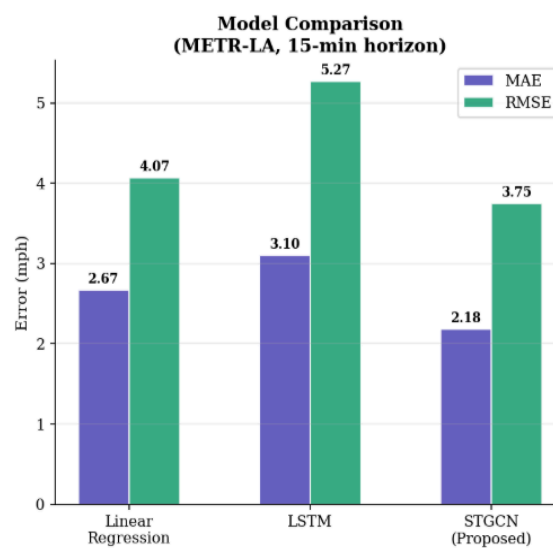


Fig. 5: Grouped bar chart comparing MAE and RMSE for all three models

C. Analysis and Discussion

As shown in Table 3, STGCN achieves the lowest error across both metrics. STGCN attains an MAE of 2.18 mph and an RMSE of 3.75 mph, representing a 29.7% MAE improvement and a 28.9% RMSE improvement over the LSTM baseline, and a 18.4% MAE improvement over Linear Regression. These improvements confirm the central hypothesis that explicitly modeling the spatial topology of the road network through graph convolution leads to consistent and measurable forecasting gains.

We can also observe that the Linear Regression (MAE 2.67) outperforms LSTM (MAE 3.10) on this dataset. This result is consistent with findings in prior literature [13] on structured highway datasets that show strong daily periodic patterns. Highway traffic follows predictable routines-rush hours at consistent times, stable free-flow speeds at night, which a linear model can exploit effectively by mapping historical observations to future values. Despite its temporal expressiveness, LSTM processes all 207 sensors as a flat input vector and must simultaneously learn complex cross-sensor dependencies and temporal patterns within a single hidden state, leading to optimization challenges. STGCN circumvents this by delegating cross-sensor communication to the graph convolution layers, allowing each component to specialize spatial structure to graph convolution, temporal dynamics to gated convolution, and achieving superior results with fewer parameters (10,609 vs. 331,599).

The error distribution analysis shown in Fig 7 shows that STGCN produces a tighter, more symmetric distribution of prediction errors centered on zero compared to LSTM, with fewer extreme outliers beyond ± 10 mph. This is particularly important for practical traffic management applications, where large and sudden prediction errors during peak congestion are more costly than small, consistent errors.

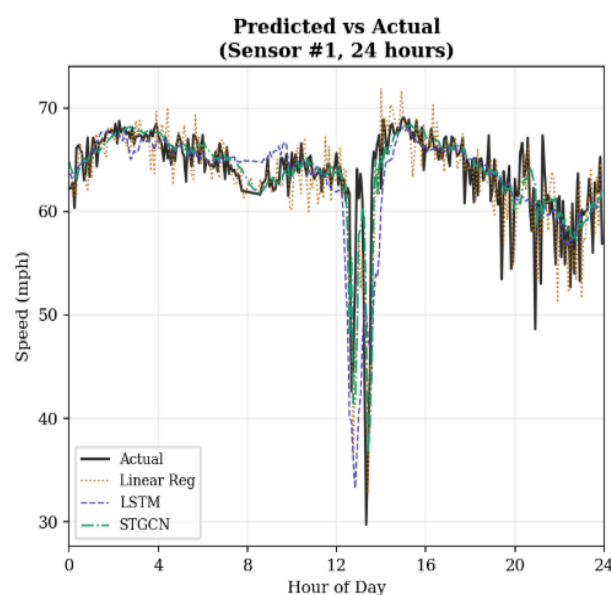


Fig. 6: Predicted vs. Actual traffic speed for all three models over a 24-hour period

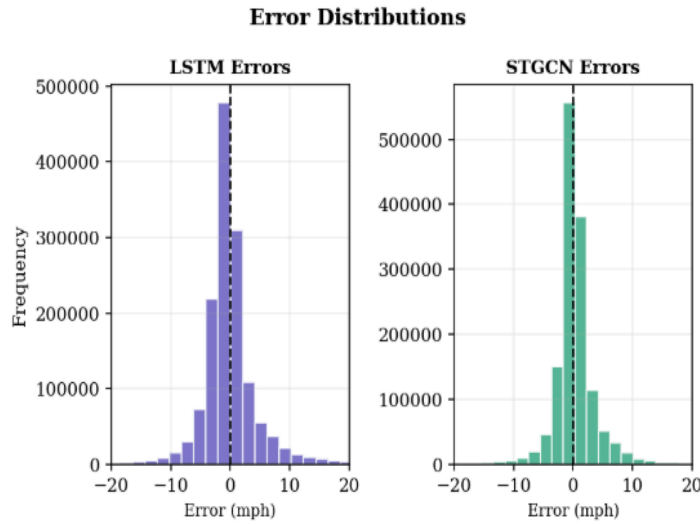


Fig. 7: Predicted error distributions for LSTM and STGCN.

V. FUTURE SCOPE IN NEPAL

Nepal's urban cities, like Kathmandu Valley, have experienced severe traffic congestion driven by rapid population growth, unplanned road expansion, and increasing vehicle registration. According to recent reports, vehicle registration in Bagmati Province exceeds 1.5 million, with annual growth rates consistently above 10% [14]. The Kathmandu road network is characterized by narrow arterial roads, a high density of uncontrolled intersections, mixed traffic flows including motorcycles, microbuses, and pedestrians, and limited grade-separated infrastructure. Key corridor junctions like Kalanki, Koteshwor, Chabahil, and Maharajgunj also experience daily peak-hour speeds dropping to near walking pace. Despite the severity of the problem, automated traffic management infrastructure is still underdeveloped. Traffic signal control at most major intersections is fixed-time or manually operated. There is no publicly available large-scale traffic speed dataset for any Nepalese city equivalent to METR-LA, making direct application of trained GNN models infeasible without dedicated data collection.

From the benchmark settings studied in this paper, we listed three challenges that distinguish the Nepali context. First, data scarcity is the most fundamental obstacle. STGCN and related models require months of dense sensor data for training. The absence of loop detectors across the Kathmandu road network means any deployment must begin with data collection. Transfer learning approaches, pre-training on METR-LA and fine-tuning on a smaller Nepal dataset, represent a promising mitigation strategy [15]. Second, road network irregularity presents difficulties for graph construction. Kathmandu's road network includes informal routes, pedestrian-vehicle mixed corridors, and segments whose effective capacity varies by

time of day and season. Third, computational infrastructure constraints may limit real-time inference deployment at traffic management centers.

Several practical pathways exist for introducing GNN-based traffic forecasting to Nepal. GPS-enabled ride-hailing services operate extensively in Kathmandu, such as Pathao and InDrive. These generate continuous spatial-temporal trajectories from which link travel speeds can be estimated via map-matching algorithms. A research partnership with such platforms could produce a Nepal-specific dataset without significant hardware investment. Alternatively, commercial traffic APIs such as HERE Traffic or TomTom, which offer free academic tiers and cover Kathmandu's major corridors at 5-minute resolution, can enable API-based data collection. OpenStreetMap provides high-quality road network data for Nepal, from which an adjacency matrix can be constructed. The Ring Road, which has a closed loop with well-defined connectivity, presents a particularly tractable initial subgraph for a first Kathmandu GNN forecasting study.

VI. CONCLUSION

This paper presents a systematic comparative study of Linear Regression, LSTM, and STGCN for short-term traffic speed forecasting on the METR-LA benchmark at a 15-minute prediction horizon. The result demonstrate that STGCN achieved the best performance with an MAE of 2.18 mph and RMSE of 3.75 mph, which outperforms LSTM by 29.7% on MAE and Linear Regression by 18.4%. Notably, STGCN achieves this performance with only 10,609 trainable parameters compared to LSTM's 331,599 parameters in the LSTM model. This demonstrates that exploiting the structural relationships of the road network can be substantially more effective than simply increasing model capacity. These results suggest that incorporating spatial topology enables more parameter-efficient traffic forecasting while maintaining superior predictive accuracy. The finding that Linear Regression outperforms LSTM highlights the importance of exploiting domain-specific structure either through periodicity (Linear Regression) or spatial topology (STGCN) rather than relying on generic temporal modeling capacity alone.

From Section V, we can identify data scarcity and road network irregularity as primary barriers in the context of Nepal and outline three practical pathways that can make a first GNN-based traffic forecasting study for Kathmandu feasible. The future work of this research will extend this comparison to multi-step horizons, explore Graph Attention Networks for adaptive spatial weighting, and conduct the GNN traffic forecasting study using a Nepal-specific dataset.

ACKNOWLEDGMENTS

The authors would like to express sincere gratitude to the faculty members of the Department of Computer Engineering, Universal Engineering and Science College, for their feedback throughout the research. The authors also acknowledge the developers of the METR-LA

dataset, which was originally compiled and released by Li et al. as part of the DCRNN project, which made this comparative study possible. Special thanks to the open-source community, whose publicly available tools and codebases supported the experimental work presented in this paper.

DATA AVAILABILITY

The METR-LA dataset used in the study is publicly available and was originally released by Li et al. [9] as part of the Diffusion Convolutional Recurrent Neural Network (DCRNN) project. No proprietary or restricted data was used in this research. All preprocessing steps, model implementations, and evaluation procedures are described in sufficient detail to allow full reproducibility of the reported results.

REFERENCES

- [1] D.Schrank, B.Eisele, and T.Lomax, "Urban Mobility Report 2019," Texas A&M Transportation Institute, College Station, TX, USA, 2019.
- [2] 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, 2014.
- [3] M.S.Ahmed and A.R. Cook, "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," *Transportation Research Record*, no. 722, pp. 1-9, 1979.
- [4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
- [5] T.N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Toulon, France, 2017.
- [6] B.M. Williams and L.A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *Journal Transp. Eng.*, vol. 129, no. 6, pp. 664-672, 2003.
- [7] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1976.
- [8] X. Ma, Z. Tao, Y. Wang, H. Yu, 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.
- [9] Y. Li, R. Yu, C. Shahabi, Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Vancouver, Canada, 2018.
- [10] B. Yu, H. Yin, Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Stockholm, Sweden, pp. 3634-3640, 2018.
- [11] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell. (IJCAI)*, Macao, China, pp. 1907-1913, 2019.

- [12] 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 Trans. Intell. Transp. Syst.*, vol. 21, pp. 3848-3858, 2020.
- [13] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883-4894, 2020.
- [14] A. Bhusal, A. Jha, A. Khanal, A. Awasthi, and A. K. Mahato, "Evaluation and Mitigation of Traffic Related Concerns in Kathmandu Valley," arXiv preprint arXiv: 2306.06121, Jun. 2023.
- [15] M. Li and Z. Zhu, "Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, pp. 4189-4196, 2021.