International Journal of Data Science
Vol. 6, No. 2, December 2025, pp. 103-112
ISSN 2722-2039

Hybrid Deep Learning for Spatiotemporal Traffic Forecasting: Integrating LSTM, Transformer, and Graph Convolutional Networks on the METR-LA Dataset

Elviyani Mawarni [a,1,*], Aria Hendrawan [a]

[a] Information Technology and Communication, Universitas Semarang, Soekarno Hatta Street Tlogosari, Indonesia
[1] elviyanim3@gmail.com
* corresponding author

ARTICLE INFO

Article history
Received April 18, 2025
Revised August 27, 2025
Accepted November 26, 2025

Keywords
Traffic forecasting
METR-LA dataset
LSTM
STGCN
TSFormer

ABSTRACT

Accurate traffic prediction in large cities such as Los Angeles is increasingly necessary as cities expand and more vehicles are added to the roads. Using the METR-LA dataset. Using the METR-LA dataset, this study proposes a hybrid deep learning architecture that combines time and space modeling techniques to improve the accuracy and scalability of traffic flow predictions. The dataset consists of multivariate time series data from 207 loop detectors that record traffic speeds every five minutes with very high resolution. This study evaluates five potential model configurations: Long Short-Term Memory (LSTM), Transformer-based TSFormer, a combination of LSTM and TSFormer, Spatio-Temporal Graph Convolutional Network (STGCN), and a model combining STGCN and TSFormer. The evaluation conducted using three performance metrics Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) were used to assess how well each model captures complex temporal and spatial relationships. Our results show that the LSTM+TSFormer hybrid model consistently outperforms all other models across all criteria. This model has the lowest MAE (0.0624) and RMSE (0.1204), meaning it is better at learning patterns that occur over time and patterns that occur rapidly. STGCN-based models are quite good at capturing spatial dependencies, but their performance improves when combined with attention-based TSFormer modules. The hybrid models introduced in this study overcome major limitations, including the narrow receptive range of recurrent networks and the inflexible spatial structures assumed in graph-based methods. This work offers important perspectives for developing forecasting models that are not only accurate and scalable but also transparent and adaptable. Future work may explore dynamic graph construction and multimodal input integration to further enhance adaptability in real-world applications.

1. Introduction

Accurate traffic forecasting has become increasingly necessary in recent years due to a sharp rise in car ownership and rapid urbanization in major cities such as Los Angeles. Traffic congestion, environmental damage, and wasted time can result when road infrastructure is unable to keep up with ever-increasing transportation demands. So, prediction models and traffic impact assessments are quite important for keeping an eye on how traffic changes and coming up with better ways to manage it. Researchers have used high-resolution traffic sensor datasets, such as METR-LA, which contains traffic speed records from 207 loop detectors across the Los Angeles freeway system and is collected every five minutes, to address the mounting challenges of urban traffic [1]. This dataset illustrates the complexity of spatial and temporal relationships in the transportation network. However, to make sense of these complex datasets, you need more advanced machine learning methods than standard visualization techniques [2]. Independent of other traffic data modeling approaches used over the years, time-series techniques, such as Long Short-Term Memory (LSTM), have been widely adopted for their ability to capture sequence dependencies[3]. On the other hand, LSTM shows weaknesses in addressing long-term dependencies and handling spatial relationships[4]. Transformer-based models were introduced to address these limitations, for example, by improving long-term sequence forecasting through self-attention mechanisms that capture global temporal patterns [5]. Next are Graph Neural Networks, which are well-suited to modeling sensor interactions across space, such as Spatio-Temporal Convolutional Graph Nets (STGCNs)[6].

Rather, however, many GNN-based techniques have static graph structures, which don't catch the dynamically evolving nature of traffic systems [7]. The proposed approach is a hybrid deep learning framework that integrates LSTM, TSFormer (an adapted Transformer architecture), and ST-GCN. We analyze five models: LSTM, TSFormer, LSTM+TSFormer, ST-GCN, and ST-GCN+TSFormer. In brief, TSFormer is designed to extract segment-level temporal representations from long historical sequences, whereas ST-GCN captures spatial correlations among sensors. By combining these advantages, the proposed model would provide better accuracy and robustness in traffic forecasting. The Spatio-Temporal Adaptive Graph Convolutional Network (STAGCN) was developed by Ma et al. [8] uses both static and adaptive graph structures to dynamically modify its adjacency matrix. Through attention mechanisms, the model discovers changes in spatial structures and operates on traffic data with its combined spatio-temporal convolutional processing layers. Testing on the PeMSD4 and PeMSD8 datasets reveals that STAGCN outperforms traditional GNN-based models in terms of RMSE and MAE.

Lin and Zha [1] developed the A3T-GCN model, which uses a Graph Neural Network architecture to evaluate performance on the METR-LA dataset. Their model achieved better forecasting results than LSTM and standard neural networks because it effectively captured spatial-temporal dependencies. Wu et al.[2] presented a model framework that uses Graph Wavelet transforms with LSTM networks to allow spatial learning through wavelet decomposition while maintaining sequential learning capabilities. The model achieved superior accuracy results on the METR-LA dataset. Zhang and Zheng[3] introduced a traffic forecasting system which disengages spatial features from temporal features to decrease representational redundancy and improve interpretability. The method successfully managed the intricate patterns of urban traffic systems.

Wang et al. [4] KGR-STGNN represents a metro traffic forecasting system that integrates knowledge graph representation learning into spatiotemporal graph neural networks. The model improved its predictive accuracy relative to LSTM and STGCN by incorporating semantic embeddings that capture external factors, such as weather and events, to enable extended prediction horizons. Oluwasanmi et al. [5] used the Knowledge Graph method with ST-GCN to incorporate contextual knowledge into the forecasting process. This integration improved explainability and forecast performance. Lei et al.[6] developed sparse Spatio-Temporal Graph Neural Networks (GCN-STGT and GAT-STGT) for traffic speed forecasting on Caltrans PeMS data. The models achieved 90% dynamic sparsity, reducing FLOPs by 10x, with only slight performance losses across short-, mid-, and long-term prediction intervals.

The Dynamic Causal Graph Convolutional Network (DCGCN) by Lin et al.[7] uses a time-varying dynamic Bayesian network (DBN) as its hyper-network to produce adaptive causal graphs that update during each time interval. The causal graphs are processed by a GCN layer to predict traffic patterns. The experimental results from METR-LA demonstrate that DCGCN achieves superior performance compared with state-of-the-art benchmarks. The Spatio-Temporal Adaptive Graph Convolutional Network (STAGCN) was developed by Ma et al.[8] uses both static and adaptive graph structures to dynamically modify its adjacency matrix. Through attention mechanisms, the model discovers changes in spatial structures and operates on traffic data with its combined spatio-temporal convolutional processing layers. Testing on the PeMSD4 and PeMSD8 datasets reveals that STAGCN outperforms traditional GNN-based models in terms of RMSE and MAE. The research team of Shin and Yoon [9] developed PGCN, a Progressive Graph Convolutional Network framework that automatically modifies graph structures during both training and inference. The construction of progressive adjacency matrices through trend similarity analysis, combined with dilated causal convolutions, gated activation units, and residual and skip connections, leads to PGCN achieving state-of-the-art performance on seven real-world traffic datasets, such as METR-LA, PEMS, and Seattle-Loop. Brimos et al. [10] demonstrated the effectiveness of combining open-government traffic data with GNN models, including TGCN and DCRNN, for traffic flow forecasting. Their evaluation shows that TGCN consistently outperformed traditional models, such as ARIMA and Historical Average, achieving up to a 70% reduction in forecasting error for short-term forecasts.

The paper by Zhong et al. [11] presents ASTG-ODE, which uses attention-based spatio-temporal Graph Neural ODEs to learn traffic dynamics via continuous-time neural ODEs combined with spatio-temporal attention. The PeMS-BAY and PeMS04 real-world datasets showed that ASTG-ODE achieves the lowest RMSE among GNN models, including DCRNN, STGCN, and STGODE, because it captures multi-horizon temporal patterns and spatial dependencies. Han et al. [12] proposed a more sophisticated spatial modelling strategy for traffic forecasting by introducing Ollivier–Ricci curvature into the message-passing procedure of spatio-temporal graph neural networks (STGNNs). Their approach effectively captures neighborhood-to-neighborhood dependencies in complex road networks, achieving significant improvements in predictive performance over traditional distance- or adjacency-based STGNNs.

Jiang et al. [13] introduced MegaCRN, a spatio-temporal forecasting model that addresses the challenges of heterogeneity and non-stationarity in traffic networks. Unlike traditional models with fixed graphs, MegaCRN incorporates a Meta-Graph Learner that generates adaptive node embeddings from a meta-memory module, enabling the model to capture diverse and evolving traffic patterns. Tested on datasets such as METR-LA, PEMS-BAY, and EXPY-TKY, MegaCRN consistently outperformed existing baselines in both accuracy and robustness, particularly under irregular traffic conditions, such as incidents and congestion. Singh et al. [14] introduced ISTGCN, which operates as a single spatio-temporal graph convolutional network to analyze spatial and temporal relationships in traffic forecasting. Their method combined block-diagonal adjacency matrices with temporal convolutions to achieve higher prediction accuracy on the PeMSD7 and PeMSD8 datasets than DCRNN and Graph WaveNet for short- and mid-term predictions.

The paper from Shao and his team in 2022[15] developed DDSTGNN, which solves the entanglement problems of traditional STGNNs by decoupling spatial and temporal dependencies. DDSTGNN enhances prediction accuracy and flexibility by creating separate models for spatial and temporal dependencies through dynamic graph construction and adaptive temporal graphs. The decoupled design of DDSTGNN outperforms DCRNN, STGCN, and MegaCRN on multiple PeMS datasets, achieving higher accuracy in tracking complex and changing traffic behavior. According to Chang et al. [16]MSSTA-GRN employs multiscale spatio-temporal modelling, in which stacked GCN layers integrate with a multiscale GRU to capture local and global traffic patterns. The authors demonstrate that their method achieves higher accuracy and greater stability than GRU, GCN, and STGCN baselines on real-world datasets across various forecasting horizons.

SeqGNN, a sequential graph neural network framework developed by Xie et al.[17], approaches traffic speed prediction through graph-to-graph sequence modeling. The combination of graph network blocks with recurrent mechanisms enables SeqGNN to capture spatio-temporal dependencies, resulting in superior performance compared to baseline Seq2Seq and RNN models for predicting dynamic urban traffic patterns. The work of Cheng et al. [18] presents AC-STSGCN as a spatio-temporal GNN that integrates traffic flow, speed, and occupancy features. The model outperformed the STGCN and DCRNN baselines on the PeMSD4 and PeMSD8 datasets, owing to its feature attention and synchronous aggregation modules.

Kong, Guo, and Liu. [19] introduced STPGNN, a GNN-based model that focuses on pivotal nodes to better capture traffic flow dynamics. By identifying and modeling these key nodes, STPGNN outperforms baselines such as DCRNN and STSGCN across multiple real-world datasets, achieving improved accuracy and efficiency. Caetano, Oliveira, and Ramos [20] conducted a comprehensive comparison of six Transformer-based models for probabilistic time series forecasting. They demonstrated that the use of explanatory variables such as price, calendar effects, and promotional events significantly improves forecast accuracy in retail demand forecasting.

Dai, Lyu, and Miao. [21] introduced FasterSTS, a lightweight spatio-temporal GCN that reduces graph computation complexity to O(KN) while effectively modeling spatial-temporal correlations. Without using RNNs or attention, it outperformed models such as STGODE and STFGNN on the PeMS datasets. Zheng et al. [22] proposed STGODE, a spatiotemporal graph neural ODE that learns continuous traffic dynamics. STGODE employed a flexible time-step prediction process and outperformed DCRNN and MTGNN on standard benchmarks.

Traffic forecasting model by Han et al. [23] integrates adaptive subgraph reformulation into a spatio-temporal deep learning framework. This model improves prediction accuracy by eliminating unnecessary inputs and generating multi-step forecasts effectively. Zhao et al. [24] developed TS-NAS, a traffic forecasting model that integrates spatial-temporal attention mechanisms with neural architecture search, and is able to flexibly select which optimal attention and convolutional modules in the architecture actually provide complementary predictive performance. TS-NAS outperformed standard models such as STGCN and ASTGCN on the PEMS datasets.

Recent research has explored diverse deep learning models for spatiotemporal traffic forecasting, including LSTM models for sequential learning, Transformer models for long-range dependencies, and Graph Neural Network (GNN) models for spatial structure modelling. Many existing approaches use these methods separately, limiting their ability to capture the complex relationships in urban traffic data. Our paper addresses this void by proposing a hybrid architecture that integrates LSTM, Transformer, and GCN modules into a single framework to improve performance, adaptability, and robustness across different spatial and temporal dynamics, using the METR-LA benchmark dataset.

2. Materials and Methods

This study aims to evaluate the performance of various deep learning architectures for multivariate time series forecasting using large-scale traffic datasets. The methodology comprises three main components: data preprocessing, model architecture implementation, and evaluation.

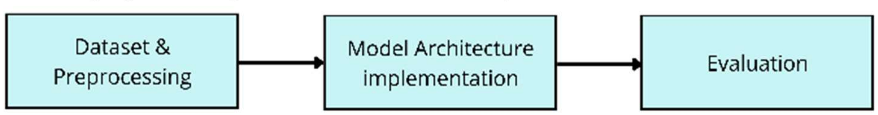

Figure 1. Flow diagram of the experimental methodology applied in this study

2.1 Dataset and Preprocessing

We employ the METR-LA dataset, which contains traffic speed recordings from 207 loop detectors in the Los Angeles highway network, collected at 5-minute intervals over four months. The dataset is provided in HDF5 (.h5) format for sensor readings and Pickle (pkl) format for the adjacency matrix that defines spatial relationships between sensors.

The preprocessing pipeline involves:

- Data Extraction: Traffic readings are retrieved from the df/block0_values key in the .h5 file, and the adjacency matrix is extracted from the .pkl file.

- Missing Value Handling: Missing or undefined values are handled using forward and backward filling or replaced with zeros.
- Normalization: Sensor data is normalized using either MinMax scaling or Z-score standardization.
- Window Segmentation: A sliding window method is applied to transform the time series into a supervised learning format. Each training sample includes 12 input time steps and 3 output time steps

The dataset is divided sequentially into training (70%), validation (15%), and testing (15%) splits to preserve temporal continuity.

2.2 Model Architectures

We implement five model configurations to capture temporal and spatial patterns in traffic flow:
LSTM: The Long Short-Term Memory (LSTM) network serves as a temporal baseline, designed to learn sequential dependencies through memory-cell updates. Its architecture typically consists of stacked LSTM layers followed by a dense output layer for final prediction.[25] At each time step t, the LSTM performs the following computations, Eq (1), Eq (2), Eq (3), Eq (4), Eq (5), Eq (6):

$$f_{\mathrm{t}} = \sigma(W_f\ [h_{\mathrm{t-1}}, x_{\mathrm{t}}] + b_f) \tag{1}$$

Eq (1) Forget Gate, this gate determines which parts of the previous cell state should be forgotten or retained.

$$i_t = \sigma(W_i[h_{\mathrm{t}}{}^{-1}, x_{\mathrm{t}}] + b_i) \tag{2}$$

Eq (2) Input Gate, the input gate decides which new information was added to the current memory cell.

$$\tilde{C}_t = tanh(W_c[h_{\mathrm{t-1}}, x_{\mathrm{t}}] + b_c) \tag{3}$$

Eq (3) Candidate Cell State. This is the candidate content that may be added to the memory state, subject to the input gate.

$$C_t = f_t \odot C_{t-1} - 1 + i_t \odot \tilde{C}_t \tag{4}$$

Eq (4) Cell State Update, The cell state is updated by combining the retained portion of the previous state and the new candidate values.

$$o_t = \sigma(W_o[h_{\mathrm{t-1}}, x_{\mathrm{t}}] + b_o) \tag{5}$$

Eq (5) Output Gate, The output gate controls how much of the current cell state is exposed as output.

$$h_t = ot \odot tanh(C_t) \tag{6}$$

Eq (6) Hidden State, The hidden state ht serves as the output of the LSTM unit at time t and is used as input to the next time step or passed to subsequent layers.

TSFormer, represents a time series variant of the Transformer structure, which uses self-attention and positional encoding to discover long-range temporal dependencies.[26] The model utilizes parallel sequence processing to generate learnable representations that predict temporal dynamics.[27] The self-attention mechanism is defined as Eq. (7):

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

Eq (7) Self-Attention, computes the relationship between different time steps by comparing query Q, key K, and value V matrices, where d_k is the key dimension.

Vol. 6, No. 2, December 2025, pp. 103-112

$$L_{MAE} = \sum_{i \in M} \|x_i - \hat{x}_i\|^2 \tag{8}$$

Eq (8) Masked Autoencoding Loss, penalizes the reconstruction error over masked positions M where x_i is the ground truth and $\hat{x}_i$ is the model's prediction.

LSTM+TSFormer, The hybrid LSTM + TSFormer model combines local temporal modeling from LSTM with global pattern recognition from TSFormer. LSTM first processes the input sequence and outputs hidden states h_t as described in Eq (1) until Eq (6). These are then passed to the TSFormer module, which applies self-attention as in Eq (7) to extract long-range dependencies. The final output is generated and optimized using Eq. (9):

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{9}$$

Eq (9) MSE Loss evaluates prediction error over the batch.

STGCN, a Spatio-Temporal Graph Convolutional Network, models spatial correlations among traffic sensors using graph convolutions, while temporal dependencies are captured using 1D convolutions along the temporal dimension. The spatial structure is described with a graph adjacency matrix, which is derived from .pkl file containing information regarding sensor connectivity. This matrix represents the relationship between nodes (sensors) based on physical road distances or some predefined topology.[28] In this case, spectral graph convolutions are used for spatial feature extraction as shown in Eq. (10):

$$H^{(l+1)} = \sigma(\widetilde{D}^{-1/2} \widetilde{A}\, \widetilde{D}^{-1/2} H^{(l)} W^{(l)} \tag{10}$$

Eq (10) Graph Convolution, combines neighboring node features using the normalized adjacency matrix $\widetilde{A}$, where $\widetilde{D}$ is the corresponding degree matrix and σ is an activation function (e.g., ReLU). Temporal dynamics are modelled through 1D convolution applied across time windows as shown in Eq. (11):

$$Y_t = \sum_{k=0}^{K-1} W_k \cdot X_{t-k} \tag{11}$$

Eq (11) Temporal Convolution captures short-term dependencies by sliding a learnable kernel $W_k$over past input sequences of length K.

STGCN+TSFormer, this enhanced model integrates TSFormer representations into the STGCN framework. Segment-level outputs from the TSFormer are fused with spatio-temporal features within STGCN, enabling the model to leverage both local and global dependencies effectively. The TSFormer module first processes the input sequence to extract long-range temporal features using the self-attention mechanism as defined in Eq. (7). The resulting representations, denoted as F_TSFormer, are then integrated with the spatial and short-term temporal features F_STGCN, which are extracted through graph convolution (Eq. 10) and temporal convolution (Eq. 11). Fusion between the two feature representations can be performed in two ways follows by Eq. (12):

$$Z = concat(F_{STGCN}, F_{TSFormer}) \tag{12}$$

Eq (12) Feature Concatenation combines spatial and temporal features from both modules into a single vector for prediction. Alternatively, weighted fusion can be applied as follows in Eq. (13):

$$Z = F_{STGCN} + \propto \cdot F_{TSFormer} \tag{13}$$

Eq (13) Weighted Fusion allows the model to balance the contribution of TSFormer by tuning the fusion parameter $\propto$. The fused representation Z is passed to a fully connected layer for final prediction.

2.3 Evaluation Metrics

The evaluation of the proposed traffic forecasting models uses three well-known error metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The test dataset serves as the basis for calculating these metrics to assess model performance on unknown datasets. Each of these metrics yields better prediction accuracy whenever the values decrease.

Mean Absolute Error (MAE) measures the average magnitude of errors between predicted and actual values, without regard to their sign.[29] It provides a straightforward interpretation of the average deviation from the true values as in Eq. (14):

$$MAE = n1i = 1\sum n \mid yi - \hat{y}i \mid \tag{14}$$

In Eq. (14), yi denotes the actual observed value, $\hat{y}i$ is the predicted value, and n is the total number of observations. This metric is less sensitive to large errors and is suitable when all individual prediction errors are equally important

Mean Squared Error (MSE): MSE calculates the average of the squared differences between actual and predicted values. By squaring the errors, MSE penalizes larger deviations more heavily, making it effective for highlighting models with large outliers. It can be calculated using Eq. (15).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2 \tag{15}$$

In Eq. (15), the squared differences amplify the impact of larger errors, allowing this metric to reflect the variance of the prediction errors. MSE is widely used when larger deviations are more critical to avoid.

Root Mean Squared Error (RMSE) is the square root of MSE and is often preferred for its interpretability, as it maintains the same units as the original data. It provides a measure of the standard deviation of the prediction errors.[30] It can be calculated using Eq. (16)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - yi)^2} \tag{16}$$

Eq. 16 demonstrates that RMSE is obtained by taking the square root of MSE, facilitating easier comparison with the original data scale. This makes RMSE especially useful for understanding the average error magnitude in practical terms.

The metric provides a clear understanding of the typical deviation from actual data points: researchers can assess model accuracy by analyzing MAE alongside MSE and RMSE, as these metrics represent overall performance (MAE), capture large errors (MSE), and provide interpretable measures (RMSE).

3. Results and Discussion

This section reports the experimental results of five deep learning architectures on the METR-LA dataset: LSTM, TSFormer, LSTM+TSFormer, STGCN, and STGCN+TSFormer. Performance evaluation is performed using three commonly-used error statistics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The main goal of this analysis is to determine which architecture has learned the most effective spatiotemporal traffic patterns.

3.1 Performance Comparison of Forecasting Models

Table 1 below summarizes the prediction performance of five deep learning models (LSTM, TSFormer, LSTM+TSFormer, STGCN, STGCN+TSFormer) evaluated using three metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE).

Table 1. Performance Comparison of Models

Model	MAE	MSE	RMSE
LSTM	0.0811	0.0251	0.1584
TSFormer	0.2741	0.3020	0.5495
LSTM+TSFormer	0.0624	0.0145	0.1204
STGCN	0.5187	0.7022	0.8380
STGCN+TSFormer	0.2807	0.3420	0.5848

Analysis of Table 1 shows that integrating different deep learning architectures significantly impacts the accuracy of traffic forecasting models. Among the evaluated models, the LSTM+TSFormer hybrid model demonstrated superior performance across all three evaluation metrics, with MAE: 0.0624, MSE: 0.0145, and RMSE: 0.1204. This indicates that the combination of temporal sequence modelling (via LSTM) and long-range attention-based learning (via Transformer) enables these predictions.

The standalone LSTM model achieves lower error than TSFormer, indicating that recurrent neural networks with memory gates are more effective at learning short- and medium-term temporal dependencies. However, when this capability is combined with the Transformer's ability to capture global contextual relationships, performance improves significantly, as observed in the hybrid configuration. The TSFormer model produces strong results, outperforming graph-based models but underperforming relative to LSTM-based models. Nevertheless, this model offers significant advantages in terms of training efficiency. The TSFormer attention mechanism enables parallel processing of sequences, unlike the sequential nature of LSTM. As a result, models that integrate TSFormer generally require shorter training times and are more scalable to large datasets.

On the other hand, the STGCN model shows the worst performance with the highest error rates across all metrics (MAE = 0.5187, RMSE = 0.8380). This suggests that, although STGCN is architecturally capable of modelling spatio-temporal dependencies through graph convolutions and 1D temporal convolutions, the model may struggle to generalize effectively in this context. However, performance improved slightly in the STGCN+TSFormer model, indicating that integrating a Transformer layer can enhance temporal pattern recognition in graph-based models, while accelerating training due to the Transformer's non-recurrent architecture.

An important factor contributing to performance variation is the difference in input data representation. The LSTM, TSFormer, and LSTM+TSFormer models rely solely on the .h5 dataset, which contains temporal traffic speed data. On the other hand, STGCN-based models use both .h5 and .pkl files, with the .pkl file encoding the spatial structure of the traffic sensor network as an adjacency matrix. While spatial information theoretically enhances model capabilities, it also increases architectural complexity and the risk of overfitting, particularly when the graph structure is suboptimal or spatial correlations are weak.

Overall, the results indicate that temporal dynamics play a more dominant role than spatial dependencies in the METR-LA dataset. As a result, hybrid models that combine temporal memory and long-range attention, such as LSTM+TSFormer, are not only the most accurate but also more computationally efficient, offering a promising balance between performance and training time for spatio-temporal traffic forecasting.

4. Conclusion

In conclusion, this study focuses on the implementation of a hybrid deep learning architecture for spatio-temporal traffic forecasting using the METR-LA dataset. By evaluating five models, LSTM, TSFormer, LSTM+TSFormer, STGCN, and STGCN+TSFormer this study highlights the effectiveness of combining temporal sequence modeling and attention mechanisms in improving prediction accuracy. Hybrid models LSTM+TSFormer, have demonstrated superior performance across all evaluated metrics, underscoring the importance of leveraging memory-based learning and global temporal context. Additionally, the addition of TSFormer not only improves model accuracy but also reduces training time due to its parallel computing capabilities. The findings of this study contribute to a better understanding of the challenges in traffic forecasting and model behavior. This reinforces the role of deep learning in addressing complex urban mobility issues by modelling traffic dynamics both in time and space. By continuing to explore dynamic spatial structures and multimodal data integration, hybrid deep learning frameworks are expected to play an essential role in developing more accurate, adaptive, and scalable traffic forecasting systems.

Acknowledgment

We thank the Information Technology and Communication Department, Universitas Semarang.

References

[1] C. Lin and Z. Zha, "Los Angeles Metropolitan Traffic Forecasting using Graph Neural Network," *2024 6th Int. Conf. Mach. Learn. Big Data Bus. Intell. MLBDBI 2024*, pp. 139–142, 2024, doi:10.1109/MLBDBI63974.2024.10823861.

[2] Q. Wu, Q. Fu, and M. Nie, "Graph Wavelet Long Short-Term Memory Neural Network: A Novel Spatial-Temporal Network for Traffic Prediction," *J. Phys. Conf. Ser.*, vol. 1549, no. 4, 2020, doi: 10.1088/1742-6596/1549/4/042070.

[3] W. Zhang and K. Zheng, "Disentangled traffic forecasting via efficient graph neural network," *J. Phys. Conf. Ser.*, vol. 2858, no. 1, 2024, doi: 10.1088/1742-6596/2858/1/012036.

[4] S. Wang, Y. Lv, Y. Peng, X. Piao, and Y. Zhang, "Metro Traffic Flow Prediction via Knowledge Graph and Spatiotemporal Graph Neural Network," *J. Adv. Transp.*, vol. 2022, 2022, doi: 10.1155/2022/2348375.

[5] A. Oluwasanmi, M. U. Aftab, Z. Qin, M. S. Sarfraz, Y. Yu, and H. T. Rauf, "Multi-Head Spatiotemporal Attention Graph Convolutional Network for Traffic Prediction," *Sensors*, vol. 23, no. 8, pp. 1–16, 2023, doi:10.3390/s23083836.

[6] B. Lei, S. Huang, C. Ding, and M. Filipovska, "Efficient Traffic State Forecasting using Spatio-Temporal Network Dependencies: A Sparse Graph Neural Network Approach," pp. 1–11, 2022, [Online]. Available: http://arxiv.org/abs/2211.03033

[7] J. Lin, Z. Li, Z. Li, L. Bai, R. Zhao, and Z. Chen, "Dynamic Causal Graph Convolutional Network for Traffic Prediction," *Proc. 2023 IEEE 3rd Int. Conf. Inf. Technol. Big Data Artif. Intell. ICIBA 2023*, pp. 229–233, 2023, doi: 10.1109/ICIBA56860.2023.10164956.

[8] Q. Ma, W. Sun, J. Gao, P. Ma, and M. Shi, "Spatio-temporal adaptive graph convolutional networks for traffic flow forecasting," *IET Intell. Transp. Syst.*, vol. 17, no. 4, pp. 691–703, 2023, doi: 10.1049/itr2.12296.

[9] Y. Shin and Y. Yoon, "PGCN: Progressive Graph Convolutional Networks for Spatial-Temporal Traffic Forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 7633–7644, 2024, doi:10.1109/TITS.2024.3349565.

[10] P. Brimos, A. Karamanou, E. Kalampokis, and K. Tarabanis, "Graph Neural Networks and Open-Government Data to Forecast Traffic Flow," *Inf.*, vol. 14, no. 4, 2023, doi: 10.3390/info14040228.

[11] W. Zhong, H. Meidani, and J. Macfarlane, "Attention-based Spatial-Temporal Graph Neural ODE for Traffic Prediction," *arXiv*, 2023, [Online]. Available: http://arxiv.org/abs/2305.00985

[12] X. Han *et al.*, "Ollivier–Ricci Curvature Based Spatio-Temporal Graph Neural Networks for Traffic Flow Forecasting," *Symmetry (Basel).*, vol. 15, no. 5, pp. 1–18, 2023, doi: 10.3390/sym15050995.

[13] R. Jiang *et al.*, "Spatio-Temporal Meta-Graph Learning for Traffic Forecasting," *Proc. 37th AAAI Conf. Artif. Intell. AAAI 2023*, vol. 37, pp. 8078–8086, 2023, doi: 10.1609/aaai.v37i7.25976.

[14] V. Singh, S. K. Sahana, and V. Bhattacharjee, "Integrated Spatio-Temporal Graph Neural Network for Traffic Forecasting," *Appl. Sci.*, vol. 14, no. 24, 2024, doi: 10.3390/app142411477.

[15] Z. Shao *et al.*, "Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting," *Proc. VLDB Endow.*, vol. 15, no. 11, pp. 2733–2746, 2022, doi: 10.14778/3551793.3551827.

[16] L. Chang *et al.*, "Multiscale Spatio-Temporal Aware Graph Recurrent Neural Network for Tra c Prediction," 2024.

[17] Z. Xie, W. Lv, S. Huang, Z. Lu, B. Du, and R. Huang, "Sequential Graph Neural Network for Urban Road Traffic Speed Prediction," *IEEE Access*, vol. 8, pp. 63349–63358, 2020, doi: 10.1109/ACCESS.2019.2915364.

[18] X. Cheng, Y. He, P. Zhang, and Y. Kang, "Traffic Flow Prediction Based on Information Aggregation and Comprehensive Temporal-Spatial Synchronous Graph Neural Network," *IEEE Access*, vol. 11, pp. 47469–47479, 2023, doi: 10.1109/access.2023.3257221.

[19] W. Kong, Z. Guo, and Y. Liu, "Spatio-Temporal Pivotal Graph Neural Networks for Traffic Flow Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 8, pp. 8627–8635, 2024, doi: 10.1609/aaai.v38i8.28707.

[20] R. Caetano, J. M. Oliveira, and P. Ramos, "Transformer-Based Models for Probabilistic Time Series Forecasting with Explanatory Variables," *Mathematics*, vol. 13, no. 5, pp. 1–29, 2025, doi: 10.3390/math13050814.

[21] B.-A. Dai, N. Lyu, and Y. Miao, "FasterSTS: A Faster Spatio-Temporal Synchronous Graph Convolutional Networks for Traffic flow Forecasting," pp. 1–13, 2025, [Online]. Available: http://arxiv.org/abs/2501.00756

[22] C. Zheng *et al.*, "Spatio-Temporal Joint Graph Convolutional Networks for Traffic Forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 372–385, 2024, doi: 10.1109/TKDE.2023.3284156.

[23] S. Y. Han, Q. W. Sun, Q. Zhao, R. Z. Han, and Y. H. Chen, "Traffic Forecasting Based on Integration of Adaptive Subgraph Reformulation and Spatio-Temporal Deep Learning Model," *Electron.*, vol. 11, no. 6, 2022, doi:10.3390/electronics11060861.

[24] C. Zhao, R. Liu, B. Su, L. Zhao, Z. Han, and W. Zheng, "Traffic Flow Prediction with Attention Mechanism Based on TS-NAS," *Sustain.*, vol. 14, no. 19, pp. 1–12, 2022, doi: 10.3390/su141912232.

[25] A. K. Takele and B. Villanyi, "Resource Aware Long Short-Term Memory Model (RALSTMM) Based On-Device Incremental Learning for Industrial Internet of Things," *IEEE Access*, vol. 11, pp. 63107–63115, 2023, doi:10.1109/access.2023.3289076.

[26] S. Liu and X. Wang, "An improved transformer based traffic flow prediction model," *Sci. Rep.*, vol. 15, no. 1, pp. 1–13, 2025, doi: 10.1038/s41598-025-92425-7.

[27] J. Lin and Q. Ren, "Rethinking Spatio-Temporal Transformer for Traffic Prediction:Multi-level Multi-view Augmented Learning Framework," 2024, [Online]. Available: http://arxiv.org/abs/2406.11921

[28] X. Chen, A. Xiao, W. Pang, X. Jiang, Z. Ren, and M. Tang, "Confidential Spatio-Temporal Graph Convolutional Networks with Differential Privacy for Traffic Forecasting," *2024 5th Int. Conf. Electron. Commun. Artif. Intell. ICECAI 2024*, pp. 66–70, 2024, doi: 10.1109/ICECAI62591.2024.10674876.

[29] H. Deng, "Traffic-Forecasting Model with Spatio-Temporal Kernel," *Electron.*, vol. 14, no. 7, 2025, doi:10.3390/electronics14071410.

[30] Z. Guo, M. Lu, and J. Han, "Temporal Graph Attention Network for Spatio-Temporal Feature Extraction in Research Topic Trend Prediction," *Mathematics*, vol. 13, no. 5, 2025, doi: 10.3390/math13050686.