Grid LSTM based Attention Modelling for Traffic Flow Prediction

Rahul Biju[1] , Sai Usha Goparaju[1] , Deepak Gangadharan[1] , Bappaditya Mandal[2]

[1]*International Institute of Information Technology, Hyderabad, India*
[2]*School of Computer Science & Mathematics, Keele University, United Kingdom*
Email: {rahul.biju, ushasai.g}@research.iiit.ac.in, deepak.g@iiit.ac.in, b.mandal@keele.ac.uk

***Abstract*—Traffic flow prediction is an important task that can directly impact the control of traffic flow positively and improve the overall traffic throughput. Although a large number of studies have been performed to improve traffic flow prediction, there are very few works on purely temporal prediction models, which is important for execution on an edge device that does not have access to spatial flow information. In order to explore the temporal prediction models further, we propose an innovative hybrid long short-term memory (LSTM) model, which we call Grid LSTM based Attention Modelling for Traffic Flow Prediction (GLSTM-A), that helps to encode temporal information better at various levels/scenarios. The proposed architecture incorporates a Grid LSTM to capture historical dependencies and a simple LSTM layer dedicated to the short-term analysis of recent data. Moreover, an innovative attention mechanism is designed to focus on the importance of data features automatically for further enhancing the model's predictive capabilities. Our proposed GLSTM-A outperforms other popular temporal prediction models such as temporal convolutional network (TCN), Bi-LSTM and LSTM, in terms of prediction accuracy and memory efficiency as mentioned in the experimental results. Experimental results and ablation studies on benchmark datasets demonstrate the superior performance of the proposed model over existing state-of-the-art models in various time series prediction tasks.**

***Index Terms*—Traffic flow prediction, Temporal Prediction, Grid LSTM, Attention Modelling, Temporal convolutional networks**

I. Introduction

The traffic conditions in several countries including India present a unique and complex challenge for traffic flow prediction due to the densely populated road conditions. Additionally, with their diverse landscape and wide range of transportation modes, Indian roads pose distinct challenges in traffic management. The coexistence of traditional means like cycles, rickshaws, and pedestrians with motorized vehicles creates intricate traffic patterns that demand customized approaches for effective control. Furthermore, the fusion of rural and urban traffic, varying infrastructure standards, and diverse vehicle driving patterns add to the complexities of Indian road traffic.

Edge computing has revolutionized processing as data can now be analyzed closer to the data source. This can be very useful for real-time traffic flow control as control decision can be performed in an edge device locally at the intersection. The flow control also benefits from real-time traffic flow prediction as flow data is available a priori. However, normally traffic flow prediction requires data to be collected from multiple neighboring intersections, which is then combined with temporal flow characteristics to perform accurate predictions. The spatial data is not always available due to unreliable communication links with neighboring intersections or the cloud. Hence, it is important to have efficient temporal prediction models that can execute on the resource constrained edge device with only local temporal flow information.

To address the above problems, this work focuses on advancing the state-of-the-art in temporal flow prediction models. Most of the temporal flow prediction models such as long short-term memory (LSTM) and temporal convolutional network (TCN), etc. primarily capture temporal features in the data, but do not extract historical periodicity data. The GLSTM-A model proposed in this work advances temporal flow prediction by integrating a parallel LSTM grid network with an attention mechanism, enabling efficient extraction of periodicity information from long-term historical data.

II. Related Work

In this section, we discuss the various approaches from literature for traffic flow prediction categorized based on the model used.

A. Machine Learning Models

Several significant contributions have advanced short-term traffic flow prediction in the field of machine learning. Notably, Castro-neto's introduction of the Online Support Vector Machine for Regression (OL-SVR) [1] marked a significant advancement, especially under atypical, non-recurring traffic conditions such as vehicular accidents and inclement weather. OL-SVR has proven to be a robust and effective tool for intelligent transportation systems, offering proactive traffic management solutions. Complementing this, a novel framework utilizing K-Nearest Neighbors (KNN) [2] was developed, tackling challenges in feature space construction and distance metric selection while also incorporating data from networked stations. Further contributions include studies by Lv et al. [3], focusing on road network traffic flows, and Duan et al. [4], who provided a comprehensive assessment of stacked autoencoders in varying traffic scenarios.

These developments, however, are surpassed by deep learning (DL) models. With their advanced neural architectures, DL models outshine traditional ML methods by offering a more nuanced, data-driven approach to traffic flow prediction, thereby providing superior accuracy and adaptability in modelling complex traffic patterns.

B. Deep Learning Models

The evolution of short-term traffic flow prediction has been markedly transformed by the advent of deep learning (DL) models, with each innovation bringing its own set of strengths and limitations. Initially, the focus was on Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, which were lauded for their proficiency in capturing both long-term and short-term temporal dependencies in traffic data, as demonstrated in studies [5], [6], [7].

However, while LSTMs excelled in temporal analysis, they fell short in addressing the spatial relationships within traffic flow data, a crucial aspect of comprehensive traffic prediction.

C. Hybrid Deep Learning Models

To bridge the gap in spatial-temporal analysis, hybrid deep learning (DL) architectures emerged, integrating Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) modules. Models such as Conv-LSTM and bi-directional LSTM [8], [9], [10] adeptly harnessed this integration, achieving a more effective capture of spatial-temporal information, a task at which traditional LSTM models fell short. These hybrid models also leveraged the periodicity of traffic flow, a crucial factor for accurate short-term prediction. By incorporating the periodic patterns into their analysis, these models could provide precise and reliable traffic flow forecasts, thus enhancing the predictability and robustness of traffic management systems [11], [12], [13]. However, the drawback is these layered and complex architectures uses the spatial information which is not always available as mentioned in the introduction. That is why we want an efficient temporal prediction model, which does not need spatial data.

The deep learning framework based on TCN, optimized using the Taguchi method by Zhao et al. [14], showed promise in improving prediction accuracy. However, these advanced models introduced complexities, leading to increased memory requirement or space complexity, demands where as, our proposed model has lower memory requirement or space complexity, due to its simple architecture, and it utilizes historical data to improve the model by extracting periodicity information, a task which TCN does not perform. Recognizing the need for optimized models, Goparaju et al. [15] developed an optimized temporal convolutional network (TCN), using a GA based approach for reduced model size and enhanced performance on edge device like RaspberryPi.

In contrast to the above works, the GLSTM-A model stands out as an efficient temporal prediction model. It surpasses the earlier described models by integrating a parallel LSTM grid, which enhances the extraction of periodic patterns in traffic data. Its use of attention modelling sharpens the focus on relevant features, leading to more accurate predictions. Demonstrating versatility, the GLSTM-A adapts effectively to various traffic scenarios in evaluations. Its s solutions in intelligent transportation systems.

In this work, we address the limitations of previous models and propose the GLSTM-A model, which offers significant advancements in short-term traffic flow prediction:

1) **Parallel LSTM grid architecture with attention modelling:** Our work introduces a new temporal flow prediction model (further referred to as GLSTM-A). It consists of a parallel LSTM grid network which extracts periodicity information from long term historical data, thereby improving the prediction accuracy over LSTM and TCN, which do not capture periodicity. This allows us to proficiently capture recurring patterns within specific time frames. As in the other temporal models, the proposed architecture also has LSTM layers, which capture the temporal information from the long term and short term data. Finally, the attention model integrated with LSTM grid emphasizes certain important features that improves the prediction accuracy further.
2) **Evaluation with two different benchmarks:** We have extensively evaluated the proposed GLSTM-A model with data from two completely different data sets - one a traffic flow data set from organized traffic scenario in PeMS dataset and another from an unorganized traffic scenario in an Indian city road. Our proposed model performed better than existing temporal models in terms of prediction accuracy and better than TCN in terms of memory resource requirements, thus making it highly suitable for deployment on resource constrained edge devices.

III. Methodology

In the context of time series analysis, several existing models, such as TCN and LSTM, have shown great promise in delivering high-performance results for various traffic flow datasets. Among these, TCN stands out due to its impressive capabilities, but it comes with a significant space complexity. Thus, the main objective of our research is to develop a novel model that not only reduces space complexity but also enhances overall performance. To achieve this, we recognize the crucial role of historical pattern analysis. By carefully considering historical patterns, our lightweight model aims to address the problem. This approach significantly enhances the accuracy of our traffic flow predictions. Importantly, our model is shown to be highly practical and efficient for real-world deployment, even in scenarios such as the Indian dataset where spatial data is unavailable.

A. Proposed Model Architecture

The proposed model adopts a dual approach when dealing with the dataset. The first approach involves historical pattern feature analysis, where we develop an innovative approach called 'Grid LSTM' to capture and analyze long-term dependencies and recurring patterns in historical data. In contrast,

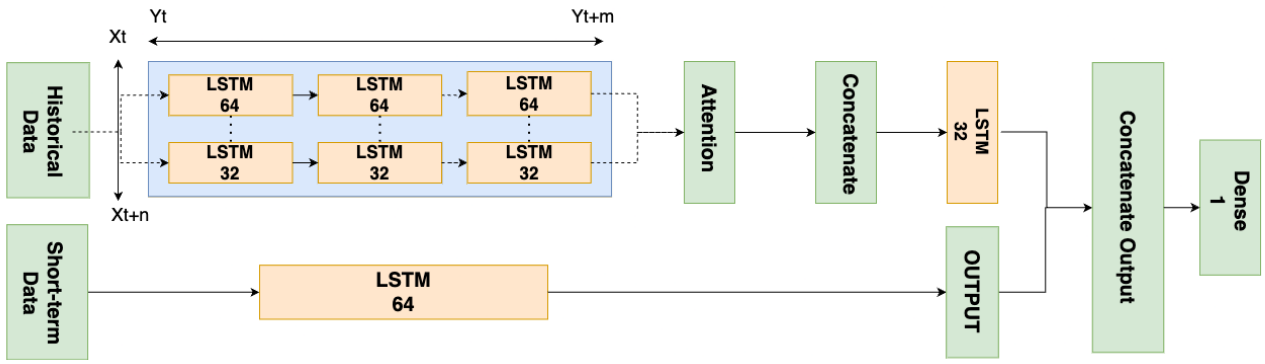

Fig. 1. Model architecture: it consists of a Grid LSTM for historical pattern analysis and a sequential LSTM for short-term prediction. The Grid LSTM uses parallel LSTM blocks to capture long-term dependencies in historical traffic data. The sequential LSTM focuses on recent data to extract short-term patterns. Our model incorporates attention mechanisms and Conv1D layers, resulting in superior predictive capabilities compared to existing TCN and LSTM models.

the traditional approach focuses on short-term pattern feature analysis, utilizing simple LSTM layers. While LSTM excels at capturing temporal dependencies in sequential data, it does have a limitation regarding its ability to retain longer memory, making it challenging to preserve prior information effectively. To overcome this limitation and further enhance the model's memory capabilities, we use Grid LSTM, which serves as a powerful reminder of prior information. This proposed model employs the Adam optimizer. The architecture is shown in Fig. 1.

The Grid LSTM plays a crucial role in analyzing traffic flow patterns at specific times on corresponding days. By considering historical data points (or prior information), the Grid LSTM facilitates a comprehensive understanding of temporal patterns across different time steps. Within the Grid LSTM block, we construct multiple sets of LSTM layers that operate in parallel. Each set of LSTM layers processes historical data from a specific date, extracting meaningful information across several time steps. Notably, the historical data points are obtained either 6 or 12 months prior to the prediction time, enabling the model to gain a comprehensive understanding of traffic patterns over a substantial period. This extensive historical perspective allows for an enhanced recognition of both regular and irregular traffic patterns, enhancing the accuracy of predictions. Additionally, the integration of data from varied historical contexts empowers the model to adapt to changing traffic conditions, making it highly effective for real-time traffic prediction.

B. Attention Mechanism

In temporal models, attention mechanisms are key for focusing on the most important time-related features. They do this by highlighting the significant parts of the data and downplaying the less important ones. The attention mechanism computes attention scores, dynamically weighing inputs based on their importance, thus allowing the model to focus on salient features. The process of computing these scores is represented in Eq. (1) and (2). The softmax-normalized scores [16], as shown in Eq. (3), ensure selective amplification of critical information, thereby enhancing the model's predictive performance and interpretability. The final attention output, which combines these weighted features with the original input, is given by Eq. (6). In these operations, $\mathbf{x}$ represents the input features, and $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ denote the weight matrices of the dense layers.

$$\mathbf{a}_1 = \mathbf{W}_1\mathbf{x} \tag{1}$$
$$\mathbf{a}_2 = \mathbf{W}_2\mathbf{x} \tag{2}$$
$$\mathbf{a}_3 = \text{softmax}(\mathbf{W}_3\mathbf{x}) \tag{3}$$
$$\mathbf{m}_1 = \mathbf{a}_2 \odot \mathbf{a}_3 \tag{4}$$
$$\mathbf{m}_2 = \mathbf{a}_1 \odot \mathbf{m}_1 \tag{5}$$
$$\mathbf{y} = \mathbf{m}_2 + \mathbf{x} \tag{6}$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{y}$ is the output of the attention layer.

C. Prior Information: Leveraging Historical Insights / Patterns

In our study, the term "prior data" refers to the dataset we collected, which spans a significantly longer period than a one month. It offers a broader view of traffic patterns, trends and changes over a more extended time frame.

The historical data matrix (H) as shown in (7) is a structured collection of past traffic flow data . It consists of rows corresponding to time steps and columns representing different days. Each element within the matrix (e.g., $x_{t_i}^{day_j}$) represents the traffic flow value observed at a specific time instant t_i on a particular day day_j. This matrix allows us to study traffic patterns and trends over time, capturing valuable insights into historical traffic behavior at distinct moments of the day across multiple days. Analyzing H enables a comprehensive understanding of the temporal dynamics of traffic flow and helps identify recurring patterns and dependencies in the data.

The present data matrix (P), as shown in (8), represents recent real-time traffic flow data observed. The historical data matrix will have data from the same time frame as the present data matrix. It consists of rows representing time steps and provides instantaneous traffic flow data. Each element (e.g.,

$x_{t_i}^{day}$) denotes the traffic flow value observed at a specific time t_i in the present moment. This matrix offers an up-to-date view of traffic conditions, essential for timely decision-making and forecasting current traffic patterns.

$$H = \begin{bmatrix} x_{t_1}^{day_1} & x_{t_1}^{day_2} & x_{t_1}^{day_3} & \dots & x_{t_1}^{day_m} \\ x_{t_2}^{day_1} & x_{t_2}^{day_2} & x_{t_2}^{day_3} & \dots & x_{t_2}^{day_m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{t_n}^{day_1} & x_{t_n}^{day_2} & x_{t_n}^{day_3} & \dots & x_{t_n}^{day_m} \end{bmatrix} \tag{7}$$

$$P = \begin{bmatrix} x_{t_1} & x_{t_2} & \dots & x_{t_n} \end{bmatrix} \tag{8}$$

In the Grid LSTM, we have 5 parallel LSTM blocks, each comprising 2 LSTM layers. The shape of the input data to Grid LSTM is represented with time-steps and the number of features or days from which data is extracted. Each set of LSTM layers receives data, allowing the model to focus on specific time-steps and extract relevant patterns. The processed tensors from these LSTM layers are appended to a list, facilitating a comprehensive analysis of various temporal aspects. Finally, these tensors are concatenated at the end. The concatenated output then passes through another simple LSTM layer to extract sequential features.

The sequential LSTM model, on the other hand, follows a more straightforward configuration, consisting of a series of LSTM layers. The input data is fed into the first LSTM layer, and its output is passed to the next LSTM layer. The output from the LSTM layers is then forwarded through a dense layer, which generates the final prediction. Its capability to capture temporal information allows it to effectively gather insights from recent data.

In development of our model, we enhanced its performance by introducing attention mechanisms to emphasize specific features and patterns. Additionally, a Conv1D layer was incorporated to improve the model's capacity in learning and extracting relevant features from input sequences. The combination of LSTM, attention mechanisms, and Conv1D layers makes our model robust and adept at handling complex time series data, such as the Indian dataset.

IV. Experiments and Results

Our work focuses on the analysis of real-world traffic data within the context of two different datasets from two diversified countries, where traffic issues vary greatly. We aim to address the challenges posed by these complex scenarios.

A. Data Collection

In our research, we employed two distinct datasets for model training: firstly, the publicly available PeMS dataset [17] and secondly, the data collected from the Hyderabad (an Indian mega-city) location coordinate with coordinates 17.444931, 78.352822 (TF-India-Hyderabad).

1) PeMS Dataset

The PeMS dataset covers a three-month period, from March 4, 2022, to June 4, 2022. It comprises 5-minute aggregate traffic data obtained from station ID 402214.

TABLE I
Impact of different time-steps on model performance using PeMS dataset.

Time-steps	RMSE	MAE	MAPE(%)
10	1.278	0.965	0.017
15	**1.155**	**0.882**	**0.0164**
20	1.141	0.899	0.0164

2) Traffic flow prediction dataset (TF-India-Hyderabad)

Our collected data spans 10 weeks, from 14^{th} December 2022, to 28^{th} February 2023. This dataset, aggregated at 15-minute intervals. It offers valuable insights into the dynamic traffic patterns specific to this dataset.

Our data preparation process involved dividing the dataset into two distinct parts: past data and present data. In past data, each input sequence comprised 15 time-steps, with each time-step representing data from a period of 5 consecutive days. For instance, a single input sequence might encompass data from December 14th to December 18th, covering 15 consecutive time-steps from 00:00 am to 01:10 am. Importantly, each time-step encapsulates traffic flow data, with a 5-minute interval between each data point. In this context, 'traffic flow' refers to the vehicular count, providing a granular view of traffic density and patterns.

The dataset is split, with 85% allocated for training and 15% for testing and evaluation. The preparation of present data involved taking 15 time-steps from a specific day. For instance, a present data sequence might consist of data from February 28th, capturing 15 consecutive time-steps from 00:00 am to 01:10 am, with a 5-minute interval between each time-step to predict the data at 01:15 am.

B. Experimental Setup

The experiments are conducted on a high-performance workstation equipped with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and a NVIDIA GeForce GTX 1650 Ti Graphics Card. For building and training the various models, we leveraged the TensorFlow-Keras framework, a popular choice for deep learning tasks. Model evaluation was carried out using two widely-used metrics: mean absolute error (MAE) and root mean square error (RMSE). These metrics provided valuable insights into the models' performance and their ability to accurately predict traffic flow and also widely used in the research community.

By utilizing these datasets and employing a rigorous data preparation process, we trained our models to effectively analyze traffic patterns and predict traffic flow. The model evaluation through MAE, MAPE and RMSE metrics ensures the reliability and accuracy of our model's predictions, providing valuable insights into the performance of the models and their ability to accurately predict traffic flow.

C. Results and Discussions

The observed superiority in performance and memory efficiency of GLSTM-A holds considerable promise for practical applications, particularly in resource-constrained environments

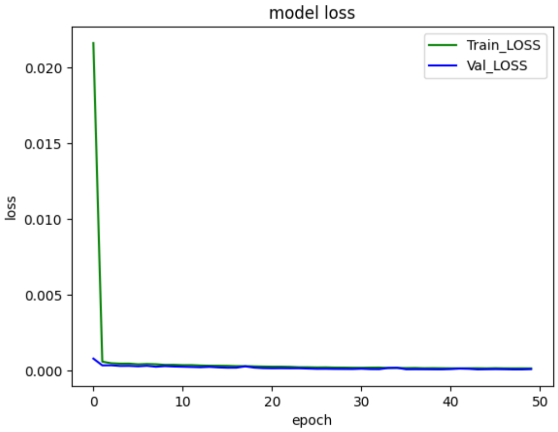

Fig. 2. Loss graph of GLSTM-A model trained using the PEMS dataset.

TABLE II
EFFECT OF DIFFERENT LSTM UNITS IN GRID LSTM ON MODEL PERFORMANCE USING PEMS DATASET.

LSTM Units in Grid LSTM	**RMSE**	**MAE**	**MAPE(%)**
32,32	1.265	0.954	0.0169
64,32	**1.155**	**0.882**	**0.0164**

TABLE III
INFLUENCE OF DIFFERENT LSTM UNITS IN LSTM LAYER ON MODEL PERFORMANCE USING PEMS DATASET.

LSTM Units in LSTM Layer	**RMSE**	**MAE**	**MAPE(%)**
32	1.1822	0.899	0.0159
64	**1.155**	**0.882**	**0.0164**
128	2.255	1.88	0.029

where careful management of memory utilization is critical. The GLSTM-A's ability to achieve lower RMSE, MAE, and MAPE values signifies its enhanced accuracy in traffic flow prediction tasks. Moreover, its memory-efficient nature ensures smoother deployment and optimized utilization of computational resources, making it an attractive and practical choice for real-world applications in traffic management and forecasting.

Fig. 2 depicts the relationship between epochs and loss for the GLSTM-A, which was trained for 50 epochs. As evident from the figure, that the model does not suffer from under-fitting or over-fitting issues, indicating a well-balanced learning process.

1) Experiments on PEMs Dataset

In our experiments, we conducted a grid search on hyper-parameters, including time-steps, LSTM units in the grid, and LSTM units in the layers. Optimal results were achieved with 15 time-steps, (64,32) LSTM units in the grid, and 64 LSTM units in the layer. Results for these parameters are presented in Tables I, II, and III. These experiments offer valuable insights into parameter choices optimizing our model's performance in time series prediction with the PeMS dataset.

TABLE IV
EXPERIMENTAL RESULTS OF GLSTM-A MODEL VARIANTS WITH ATTENTION, CONVOLUTION AND ADDITIONAL LSTM LAYERS ON PEMS DATASET.

Model	RMSE	MAE	MAPE
GLSTM-A (attention both input)	94.679	65.376	0.201
GLSTM-A (attention hist. input)	89.42	55.497	0.14
GLSTM-A (adding lstm to Grid LSTM)	90.667	52.70	0.124
GLSTM-A (adding conv layer to Grid LSTM)	**88.47**	**51.988**	**0.124**

In our comprehensive evaluation of LSTM, TCN, and our proposed GLSTM-A, the GLSTM-A demonstrated exceptional performance, significantly outperforming other models with lower values of root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) as presented in Table V. In contrast, CNN-LSTM and Conv-LSTM models are adept at integrating both spatial and temporal features, effectively capturing interdependencies across interconnected stations. Although incorporating spatial features offers potential advantages in capturing intricate patterns.

2) Experiment on PeMS Dataset Across Different Stations

The Table VII presents a comprehensive comparison of performance metrics across three distinct models TCN, LSTM, and GLSTM-A evaluated on different datasets associated with specific station IDs.

When focusing on the MAPE metric, the GLSTM-A model demonstrates superior performance for stations 402214 and 402835, with the lowest MAPE values of 0.016 and 0.018, respectively. However, for station 414025, both the TCN and GLSTM-A models are fairly comparable, with MAPE values of 0.208 and 0.206, respectively. On the other hand, the LSTM model consistently shows higher MAPE values across all stations, indicating its lesser accuracy in predictions in this specific context. In terms of performance Across Stations, Station 402835 seems to be a challenging dataset for all the models, with higher error values across all metrics when compared to station 402214. This could be indicative of inherent complexities or variances within the data from this station. With respect to consistency across Metrics, the GLSTM-A model not only excels in MAPE but also performs remarkably well in RMSE and MAE metrics for stations 402214 and 402835. However, for station 414025, its RMSE and MAE values are relatively close to those of the TCN and LSTM models. This suggests that while GLSTM-A might be well-suited for certain datasets, its performance might converge with other models for more complex or varied datasets.

3) Experiment on TF-India-Hyderabad Dataset

The TCN architecture's inherent complexity demands greater memory resources. To further clarify this, we conducted an in-depth analysis of memory consumption, as noted in Table VI. Our findings revealed that the TCN model

TABLE V
COMPARATIVE ANALYSIS OF MODEL PERFORMANCE USING PEMS DATASET.

Model	RMSE	MAE	MAPE(%)
LSTM [5]	3.428	2.446	0.0452
TCN [14]	2.186	0.899	0.0168
Bi-LSTM [18]	3.118	2.312	0.050
GLSTM-A	**1.155**	**0.882**	**0.0164**

TABLE VI
MEMORY REQUIREMENTS AND PREDICTION TIME COMPARISON BETWEEN GLSTM-A AND OTHER MODELS TRAINED USING THE TF-INDIA-HYDERABAD DATASET.

Model	Memory Required (MB)	Prediction Time(ms)
TCN [14]	324.98	3
GLSTM-A	**132.96**	11
Bi-LSTM [18]	35.14	6
LSTM [5]	15.14	1.6

requires the highest memory resources, reaching a substantial 324.98 MB. In stark contrast, our proposed GLSTM-A, which primarily leverages temporal features, showcased remarkable memory efficiency, consuming only 132.96 MB. Our model takes 0.00110s for one prediction. The Bi-LSTM model, which processes time series data in both forward and backward directions, also demonstrated a moderate memory footprint of 35.14 MB. This pronounced disparity in memory usage underscores the inherent complexity of the TCN architecture compared to the more streamlined design of the GLSTM-A.

In Table IV, we noted an experimental evaluation of various GLSTM-A model variants with distinct configurations, including attention mechanisms, convolutional layers, and additional LSTM layers. The convolutional layer is added within the grid LSTM block as one of the layers. The results, as indicated by the performance metrics, showcase the impact of these variations on the model's predictive accuracy. For instance, the GLSTM-A model with attention applied to both input types achieved an RMSE of 94.679, MAE of 65.376, and MAPE of 0.201. Comparatively, the variant with attention applied to historical input demonstrated improved performance with an RMSE of 89.42, MAE of 55.497, and MAPE of 0.14. Furthermore, the inclusion of an additional LSTM layer to the Grid LSTM resulted in an RMSE of 90.667, MAE of 52.70, and MAPE of 0.124, while the introduction of a convolutional layer to the Grid LSTM yielded an RMSE of 88.47, MAE of 51.988, and MAPE of 0.124. The Bi-LSTM variant, which incorporates bidirectional processing of time series data, showed a further enhanced capability with an RMSE of 3.118, MAE of 2.312, and a MAPE of 0.050, revealing the bidirectional approach's strength in capturing temporal dependencies.

Additionally, spatial models such as CNN-LSTM [11] and Conv-LSTM [8] were also evaluated. The CNN-LSTM model registered an RMSE of 1.77, MAE of 5.19, and MAPE of 0.0159, while the Conv-LSTM model achieved an RMSE of 1.714, MAE of 4.960, and MAPE of 0.0164. Despite

TABLE VII
EXPERIMENTS ON DIFFERENT MODELS ACROSS VARIOUS STATIONS IN THE PEMS DATASET.

Model	Station Id	MAPE	RMSE	MAE
TCN [14]	402214	0.0168	2.186	0.899
	402835	0.50	8.754	5.960
	414025	0.208	8.927	6.460
LSTM [5]	402214	0.0452	3.428	2.446
	402835	0.478	8.631	5.814
	414025	0.233	8.954	6.368
GLSTM-A	**402214**	**0.016**	**1.155**	**0.882**
	402835	**0.018**	**1.519**	**1.13**
	414025	**0.206**	**8.125**	**5.9**

the strengths of spatial feature extraction inherent in these models, the proposed GLSTM-A model demonstrated superior performance, affirming that it better captures the complexities of the dataset. These comparisons highlight the proposed model's robustness and its superior ability to model temporal dynamics when compared to models that primarily focus on spatial features.

D. Ablation Study

In the ablation study, we conducted experiments to investigate the impact of the number of layers in the Grid LSTM of our proposed model. The objective was to understand how the depth of the Grid LSTM influences the predictive performance of the model. We conducted experiments with different configurations, ranging from 1 layer to 8 layers, and evaluated the model's performance using RMSE, MAE, and MAPE.

The results of the ablation study are summarized in Table VIII. As we increased the number of layers in the Grid LSTM, we observed a consistent decrease in the error values. Specifically, as the number of layers increased from 1 to 5, both RMSE and MAE showed a notable reduction. This indicates that adding more layers to the Grid LSTM led to improved predictive performance, as it allowed the model to capture more intricate temporal dependencies in the traffic flow data. Interestingly, beyond 5 layers, the model's RMSE ,MAPE and MAE values reached a plateau, suggesting that further increasing the number of layers did not result in significant performance improvements. This finding indicates that a certain depth of the Grid LSTM was sufficient to capture the essential temporal patterns in the data, and adding more layers did not provide substantial benefits in terms of predictive accuracy.

The choice of the number of LSTM layers should be made based on the characteristics and requirements of the specific dataset in use. When the Grid LSTM component is removed, the model essentially becomes a simple LSTM model. As a result, the loss metrics are expected to be higher for the simple LSTM model compared to GLSTM-A as shown in table V.

Table IX summarizes the performance of various Grid LSTM configurations, using the TF-India-Hyderabad dataset. The results indicate that the 5-layer LSTM configuration yields

TABLE VIII
PERFORMANCE METRICS FOR DIFFERENT NUMBER OF LSTM PARALLEL LAYERS IN GRID LSTM CONFIGURATIONS WITHIN GLSTM-A USING THE PEMS DATASET.

GLSTM-A	RMSE	MAE	MAPE
1 layer	1.62	1.21	0.019
3 layer	1.39	1.20	0.018
5 layer	**1.15**	**0.88**	**0.016**
6 layer	1.28	0.98	0.019
8 layer	1.29	0.98	0.019

TABLE IX
PERFORMANCE METRICS FOR DIFFERENT NUMBER OF LSTM PARALLEL LAYERS IN GRID LSTM CONFIGURATIONS WITHIN GLSTM-A USING THE TF-INDIA-HYDERABAD DATASET.

GLSTM-A	RMSE	MAE	MAPE
1 layer	94.94	55.58	0.130
3 layers	92.17	56.18	0.140
5 layer	**88.47**	**51.988**	**0.124**
6 layers	90.67	54.50	0.136
8 layers	92.75	54.84	0.132

the most favorable outcomes with the lowest RMSE (88.47), MAE (51.988), and MAPE (0.124). However, increasing the number of layers beyond five does not lead to consistent improvements, suggesting an optimal layer configuration for this dataset and emphasizing the criticality of model depth in deep learning-based time series forecasting.

V. CONCLUSION AND FUTURE WORK

In this work we have addressed the challenges involved in traffic flow prediction, which is of paramount importance for overall safety of any transportation system or traffic network. To address one of the existing challenges, such as computational efficiency and predictive performance and encoding temporal information, we proposed GLSTM-A, that comprises of an innovative hybrid long short-term memory model that uses historical or prior information for encoding crucial temporal information at multiple levels/scenarios. We also proposed an innovative attention mechanism to focus on specific region of the data features automatically for further improving the model's predictive capabilities. It's notable that the model was trained on two different datasets from distinct locations or countries, where traffic issues vary greatly. Despite these differences, our method demonstrated strong performance across all tested cases. Extensive experimental results and ablation studies demonstrated that our GLSTM-A architecture outperforms popular LSTM,CNN-LSTM,Conv-LSTM and TCN models, all while maintaining reduced computational complexity and memory usage for various time series prediction tasks. However, it is important to note a limitation of the model. GLSTM-A may not perform optimally in scenarios with limited training data. When there is a scarcity of data for training, the model's predictive performance may be adversely affected.

For future endeavors, we aim to further augment the capabilities of our model, particularly in long-term prediction tasks. We are aiming to integrate spatial information into our model, making it more comprehensive and robust. By including spatial features, the model will attain a deeper understanding of traffic flow patterns and trends, enabling even more accurate and reliable short-term and long-term predictions.

REFERENCES

[1] Castro-Neto *et al.*, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert systems with applications*, vol. 36, no. 3, pp. 6164–6173, 2009.

[2] H. Hong, W. Huang, X. Zhou, S. Du, K. Bian, and K. Xie, "Short-term traffic flow forecasting: Multi-metric knn with related station discovery," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 1670–1675, IEEE, 2015.

[3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[4] Y. Duan, Y. Lv, and F.-Y. Wang, "Performance evaluation of the deep learning approach for traffic flow prediction at different times," in *2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 223–227, IEEE, 2016.

[5] P. Poonia and V. K. Jain, "Short-term traffic flow prediction: Using lstm," in *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*, pp. 1–4, 2020.

[6] V. Osipov, V. Nikiforov, N. Zhukova, and D. Miloserdov, "Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers," *Neural Computing and Applications*, vol. 32, no. 18, pp. 14885–14897, 2020.

[7] B. Yang, S. Sun, J. Li, X. Lin, and Y. Tian, "Traffic flow prediction using lstm with feature enhancement," *Neurocomputing*, vol. 332, pp. 320–327, 2019.

[8] Z. Lu, J. Xia, M. Wang, Q. Nie, and J. Ou, "Short-term traffic flow forecasting via multi-regime modeling and ensemble learning," *Applied Sciences*, vol. 10, no. 1, p. 356, 2020.

[9] Zheng *et al.*, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Transactions on ITS*, 2020.

[10] Q. Li, H. Tan, Y. Wu, L. Ye, and F. Ding, "Traffic flow prediction with missing data imputed by tensor completion methods," *IEEE Access*, vol. 8, pp. 63188–63201, 2020.

[11] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016.

[12] Lee *et al.*, "Comparative analysis of machine learning algorithms to urban traffic prediction," in *ICTC 2017*, IEEE.

[13] Wu *et al.*, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, 2018.

[14] Zhao *et al.*, "Deep temporal convolutional networks for short-term traffic flow forecasting," *IEEE Access*, vol. 7, pp. 114496–114507, 2019.

[15] S. U. Goparaju, R. Biju, M. Pravalika, M. Bhavana, D. Gangadharan, B. Mandal, and C. Pradeep, "Optimization and performance evaluation of hybrid deep learning models for traffic flow prediction," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–7, IEEE, 2023.

[16] S. Totaro, A. Hussain, and S. Scardapane, "A non-parametric softmax for improving neural attention in time-series forecasting," *Neurocomputing*, vol. 381, pp. 177–185, 2020.

[17] "California Department of Transportation, USA, Feb 2022." https://pems.dot.ca.gov/.

[18] J. Wang, F. Hu, and L. Li, "Deep bi-directional long short-term memory model for short-term traffic flow prediction," in *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part V 24*, pp. 306–316, Springer, 2017.