Long-Term Traffic Prediction Based on Stacked GCN Model

Atkia Akila Karim [1,*], Naushin Nower [2]

Institute of Information and Technology, University of Dhaka,
Suhrawardi Udyan Rd, Dhaka 1200, Bangladesh
[1] msse1760@iit.du.ac.bd; [2] naushin@iit.du.ac.bd*
* corresponding author

ARTICLE INFO	ABSTRACT
	With the recent surge in road traffic within major cities, the need for both short and long-term traffic flow forecasting has become paramount for city authorities. Previous research efforts have predominantly focused on short-term traffic flow estimations for specific road segments and paths. However, applications of paramount importance, such as traffic management and schedule routing planning, demand a deep understanding of long-term traffic flow predictions. However, due to the intricate interplay of underlying factors, there exists a scarcity of studies dedicated to long-term traffic prediction. Previous research has also highlighted the challenge of lower accuracy in long-term predictions owing to error propagation within the model. This model effectively combines Graph Convolutional Network (GCN) capacity to extract spatial characteristics from the road network with the stacked GCN aptitude for capturing temporal context. Our developed model is subsequently employed for traffic flow forecasting within urban road networks. We rigorously compare our method against baseline techniques using two real-world datasets. Our approach significantly reduces prediction errors by 40% to 60% compared to other methods. The experimental results underscore our model's ability to uncover spatiotemporal dependencies within traffic data and its superior predictive performance over baseline models using real-world traffic datasets.

I. Introduction

Traffic flow prediction is a crucial research domain focused on anticipating forthcoming traffic patterns within a road network [1]. Recently, this field has garnered increasing interest due to the rapid advancements and adoption of Intelligent Transportation Systems (ITS). Traffic flow prediction plays a fundamental role within the framework of ITS, serving as a pivotal component, plays a critical role in traffic management and planning, and aims to provide better transport management by avoiding congestion.

In most megacities, traffic congestion is a significant issue that hinders residents' daily lives and the nation's economic progress [2]. The significant causes of traffic congestion include rising population, urbanization, poor traffic management, and inadequate transportation infrastructure [3]. The economic burden of traffic congestion in urban centers is steadily increasing globally, affecting nearly every major city. For instance, in Dhaka, traffic congestion results in the loss of five million working hours daily, translating to an annual economic toll ranging from 200 to 550 billion takas [4]. Such severe traffic congestion can harm a nation's economy, hinder foreign investments, disrupt the supply and demand dynamics, and contribute to heightened emotional stress among the population [5]. Consequently, timely and precise traffic flow forecasting is immensely valuable to urban residents. Travelers can create better trip arrangements with accurate traffic flow forecasting, reducing traffic congestion, fuel consumption, and carbon emissions [6].

However, because of its intricate spatial and temporal connections and abrupt accidents, traffic flow prediction has always been a complex problem. Numerous specialists and academics have dedicated their research to studying traffic flow prediction and have developed numerous prediction

techniques that can be categorized depending on the model: parametric or nonparametric. Parametric models derive their parameters by analyzing the original data, and traffic forecasts are subsequently executed based on predefined regression functions. Various traditional parametric models like the ARIMA model [7], the KF model [8][9], and different variations of ARIMA have been utilized for traffic flow prediction. Nevertheless, due to traffic flow's nonlinear and stochastic nature, these models often struggle to provide accurate predictions.

Consequently, nonparametric models, including random forest [10], support vector machine [11][12], fuzzy logic models [13], Bayesian networks [14], K-Nearest Neighbors methods [15][16], neural network models [17][18], and hybrid combinations of these algorithms, have been introduced. These models can handle spatiotemporal data, although their effectiveness may vary depending on the application and dataset size. Despite their superior performance, these models encounter challenges when dealing with extensive traffic datasets.

To address these challenges, recent advancements in deep learning networks have become increasingly prevalent, as they can handle large datasets and improve prediction accuracy by utilizing multiple layers to extract intricate traffic characteristics. For instance, Wu and Tan [19] introduced a model featuring a one-dimensional Convolutional Neural Network (CNN) for capturing spatial features and incorporated two Long Short-term Memory (LSTM) layers to capture temporal patterns. Duan et al. [20] adopted CNN for spatial features and combined it with LSTM for temporal feature extraction. Additionally, they employed a greedy training policy to reduce training time and enhance accuracy, especially in deeper networks. However, CNN has inherent limitations when dealing with complex topological structures, as it was initially designed for Euclidean spaces like images and regular grids, making it less suitable for adequately characterizing the spatial intricacies and dependencies within road networks. The Graph Convolutional Network (GCN) [21] was introduced to address this limitation. GCN represents the traffic network as a graph and effectively captures spatial attributes from neighboring nodes. In another study [22], a combination of GCN was utilized for traffic flow prediction, incorporating LSTM and multitask learning to capture global and local traffic flow correlations along road segments. This model leveraged GCN within an undirected graph framework to depict the spatial distribution patterns of taxi trips and used LSTMs to capture temporal features.

Additionally, the implementation of multitask learning enhanced the model's generalizability. In [23], an approach called Hierarchical Graph Convolution Networks (HGCN) was proposed, operating on both micro and macro traffic graphs. This study recognized the hierarchical structure of traffic systems, comprising microlayers (road networks) and macro layers (region networks). In [24], the authors emphasized the importance of learning node-specific patterns without relying on predefined graphs. To achieve this, they introduced two adaptive modules: the Node Adaptive Parameter Learning (NAPL) module, capturing node-specific patterns, and the Data Adaptive Graph Generation (DAGG) module, inferring interdependencies among traffic series automatically. These modules were integrated with recurrent networks to create the Adaptive Graph Convolutional Recurrent Network (AGCRN), effectively capturing fine-grained spatial and temporal correlations in traffic data. However, it is worth noting that these innovative methods predominantly focused on short-term traffic prediction despite the increased complexity associated with long-term prediction. Long-term traffic prediction is particularly challenging due to its essential applications in traffic management and schedule routing planning. Consequently, research is scarce in this domain, primarily because predicting the distant future presents more considerable difficulties compared to short-term forecasting.

Long-term traffic flow prediction is a less frequently explored research area, and achieving accurate long-term predictions poses challenges due to performance degradation over extended timeframes compared to short-term predictions. A previous study [25] employed a Recurrent Neural Network (RNN) with GPU acceleration to forecast long-term traffic flow in Odense and Beijing. However, it is worth noting that RNNs are susceptible to the vanishing gradient problem, which can impact their performance. In another study [26], a spatial-temporal graph attention network was introduced, designed to capture the data's dynamic graph structure and spatial-temporal dependencies. Their model is tested using two public datasets gathered in California. In their study,

Wang et al. [27] introduced a deep learning architecture comprising two main components: a bottom-up LSTM encoder-decoder structure and a top-down calibration layer.

On the other hand, Li et al. [28] proposed a hybrid model for forecasting next-day traffic flow. This model incorporates wavelet decomposition, CNN, and LSTM techniques. In [29], CNN and BiLSTM are incorporated to predict long-term traffic flow. However, CNN is unsuitable for capturing the complex traffic road network structure since it is based on Euclidean distance. Moreover, as those prediction techniques do not use separate models, errors can propagate quickly, and those models find difficulties in handling sudden incidents. Accurately predicting traffic patterns beyond short time frames remains challenging due to the inherent complexities of error accumulation in existing models, which undermines long-term forecasting precision. To solve those problems, we proposed a stacked GCN that can handle sudden incidents, and as there is a GCN for every segment, the error does not propagate. Most models use RNN or its variant to capture the temporal feature and CNN or GCN to capture the spatial feature. However, using separate models has drawbacks; it cannot capture the inherent interrelationship between temporal and spatial features. To overcome this, we used stacked GCN, where segmented modules inherit the temporal feature that helps GCN capture both the spatial and temporal features simultaneously.

In the proposed architecture, we design a segmented module that segments input data to extract the temporal features and then incorporates a GCN for every segment to give day-long predictions. Thus, we use stacked GCN to get the final prediction outcome based on the segment, and as a result, because of stacked architecture, the error from the previous outcome is not propagated in the next prediction. GCN is utilized in the proposed method since it improves CNN, which can directly handle graphs and non-Euclidian distance and thus works better in road networks. Our contributions to this paper are briefly summarized below:

- We proposed a stacked GCN predictive model for traffic flow over extended periods and applied segments to improve the prediction performance without accumulating errors.
- We used two publicly available datasets to evaluate our model and perform a whole-day prediction. We conducted a comparative analysis of our model against the baseline methods, and our model shows superiority in traffic forecasting.

II. Method

This section introduces the proposed Stacked GCN model designed for long-term traffic flow prediction. Our architecture leverages GCN to extract intricate spatial relationships within the road network. The road network, represented as the graph $G = (V, E)$, serves as the input to GCN, encapsulating the topological structure of the road network. Each road is treated as a node, illustrated in Figure 1, and the edges denote connections between the roads.
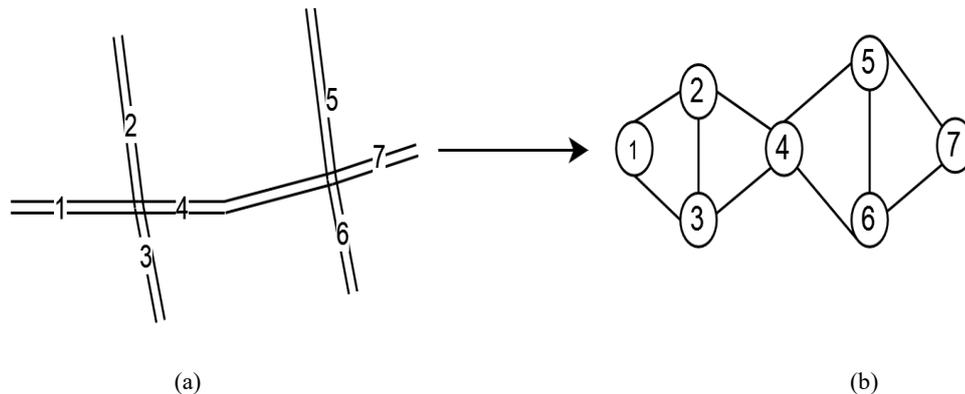

| (a) | (b) |

Fig. 1. Real road structure transformation into graph road network where (a) Road map (b) Graph structure of the road map

Within the graph, individual roads are node representations, with V being the set of road nodes $V = \{v_1, v_2, \cdots, v_N\}$, N signifying the total number of nodes, and E representing the set of edges. The adjacency matrix $A \in R^{N \times N}$ characterizes road linkages, with entries in the matrix being 0 for unrelated roads and 1 for connected ones. The feature matrix $X \in R^{N \times F}$, with F corresponding to the

historical traffic flow data length. Our primary objective is to predict traffic flow for the next T time steps, relying on historical data. The proposed Stacked GCN model comprises two essential modules: i) a segmented module and ii) a graph convolutional network module. GCN effectively captures spatial traffic data characteristics, while the segments module divides historical data into S segments, enabling the model to learn temporal patterns. The primary goal of our suggested model is to create a more accurate forecast, and the divergence from the actual value should be minimized. As a result, our goal is to reduce prediction error, which can be expressed as in (1).

$$min \left(||Y_i - Y_{-i}|| \right) \qquad\qquad (1)$$

Y_i represents the actual observed value of traffic flow, while Y_{-i} signifies the predicted output. The methods of the modules are described in the following subsections.

A. Segmented Module

To capture the periodic information embedded within the historical data, we employ the segmented module, which transforms the full-length historical traffic data (X) into a collection of periodic segments denoted as $S = \{S1, S2, \ldots, Sd\}$, where d represents the number of segments. Each of these segments encapsulates historical data from a distinct period, with Si representing a sub-time series conveying information about a specific period. Here, l signifies the length of each segment, and Si is composed of temporal features about the corresponding time interval. Figure 2 illustrates an illustrative example of this data segmentation process, where the previous four days' twenty-four-hour data is segmented into six segments. Each segment consists of four hours. So, the value of d is six, and the value of l is four hours. The fifth day's data were predicted using the previous four days' data segments.
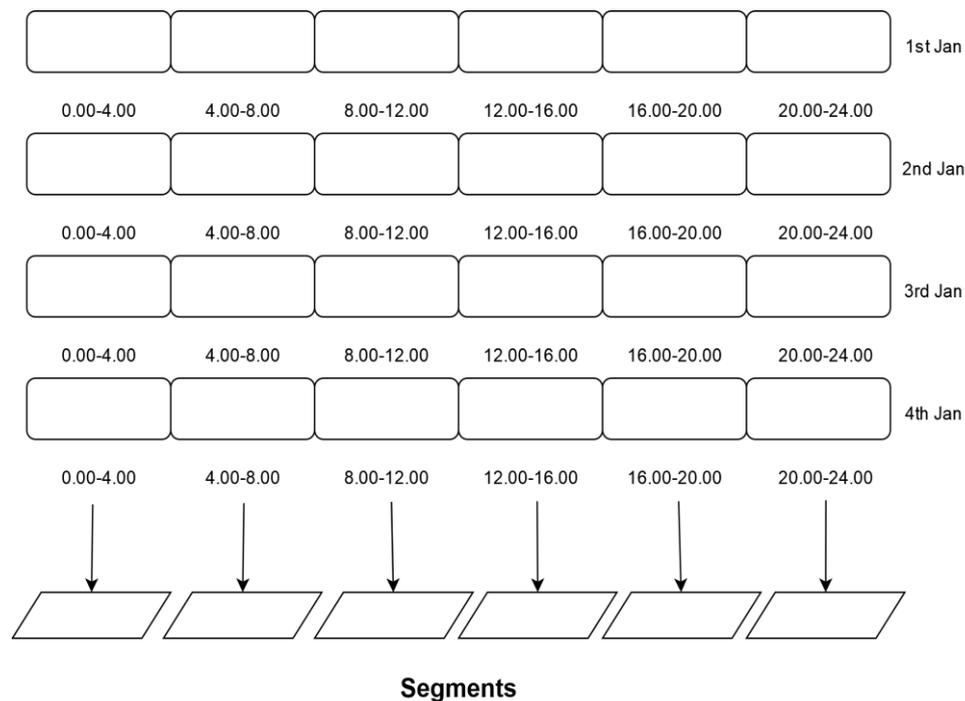

Fig. 2. Segmentation mechanism of input data

In this proposed method to predict a time stamp, we have considered the same time segment from the historical data rather than the whole historical data. Typically, traffic behavior within a region exhibits a consistent pattern during the same periods across different days. As a result, historical daily patterns can be characterized as recurring weekly patterns within specific time windows. For instance, the traffic speed observed on a Wednesday at 8:00 AM and 9:00 AM will resemble the corresponding time slots on previous days. Consequently, the repetitive patterns in traffic data from preceding days within a specific time window can serve as a valuable reflection of the historical daily

trends. Thus, we have extracted the temporal features from the segmented module from the historical data, and from the stacked GCN module, we have considered the traffic speed for that particular time segment.

B. Graph Convolutional Networks

The GCN model collects spatial features from its first-order neighborhood. As depicted in Figure 3, node *a* represents a central road, while nodes *b* and *c* signify the roads connected to this central road. Spatial features are extracted by establishing the topological relationships between the central and neighboring roads. The GCN model generates a Fourier domain filter utilizing the adjacency matrix 'A' and the feature matrix 'X.' This filter, applied to the nodes within the graph, gathers spatial characteristics from the first-order neighborhood of each node. The GCN model is constructed by stacking multiple convolutional layers, allowing it to capture increasingly complex spatial relationships among the nodes, as in (2).

$$H^{l+1} = \sigma \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \tag{2}$$

$H^{(l)}$ represents the node feature matrix at layer l, $A = A + I$ is the adjacency matrix of the graph with self-connections added, D is the degree matrix, W(l) denotes the learnable weight matrix at layer l, and σ represents a nonlinear activation function. The number of layers in the model determines the maximum distance over which node characteristics can propagate and interact within the graph structure. With one layer GCN, for instance, each node can only obtain information from its neighbors. Each node's information-gathering operation runs simultaneously and independently. We repeat the process of obtaining information when we layer another layer on top of the original one. However, GCN suffers from a vanishing gradient problem if more layers are added, precisely more than four layers, causing limited performance [30]. To avoid this problem, we used two layers in GCN that can better handle non-euclidean road networks compared to CNN without suffering from the vanishing gradients problem. We utilize historical traffic data as our input, segment the input data, and then use a two-layered Graph Convolution Network (GCN) for every segment.
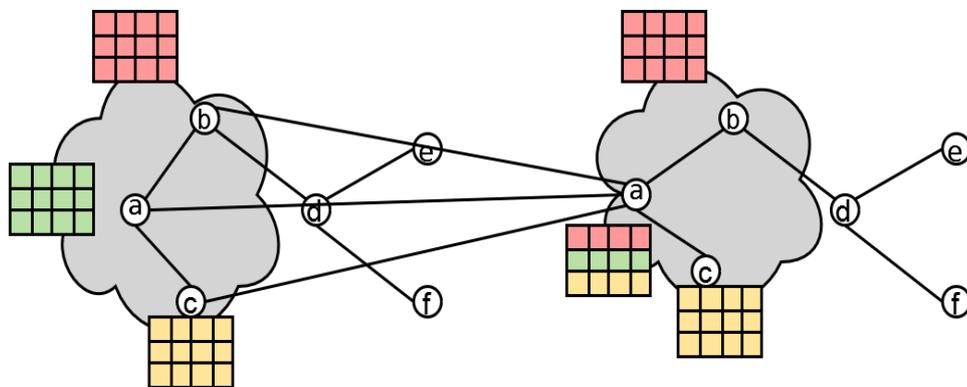

Fig. 3. Graph Convolutional Network (GCN)

C. The Proposed Stacked-based GCN Model

The architecture of our proposed model, as depicted in Figure 4, incorporates a segmented module responsible for preprocessing the input time series data X and converting it into periodic segments denoted as *S*. For day-long prediction, we have segmented twenty-four hours into *S* segments. We have generated results for different numbers of segments.
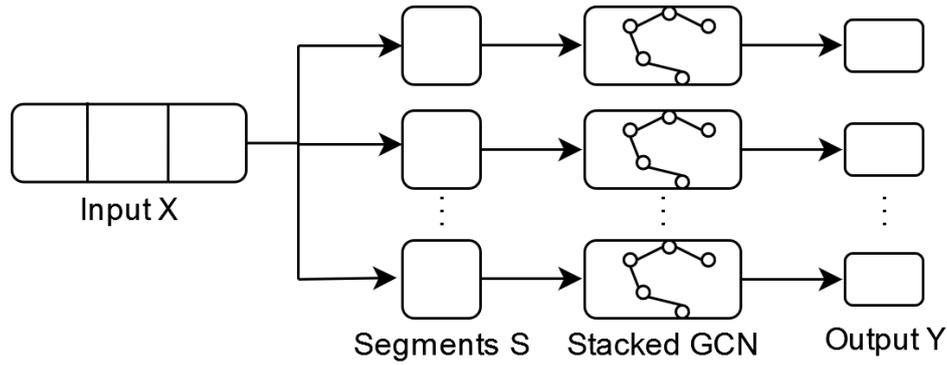
Fig. 4. Graph Convolutional Network (GCN)

Table 1 demonstrates that an increase in the number of segments leads to reduced error. Twenty-four segments give less error than others (2, 3, 4, 6, 8, 12). In twenty-four segments, each segment consists of one-hour timestamps. In the GCN model, the processed segments are utilized to generate the final predictions for traffic speed data. As depicted in Figure 1, the initial raw historical data is initially input into the system, and from there, the segments are extracted for further processing. In Fig. 3, we have demonstrated the segmentation of historical data for our model. Previous four days, particular segments (suppose 4:00 PM - 5:00 PM) have been considered to predict the fifth day's 4:00 PM - 5:00 PM. Every day is divided into S segments. After that, in the stacked GCN models, GCN models are used to process each segment separately. Every GCN used for the segment is two-layered. The outputs of these modules are then merged to produce the final prediction sequence Y. Historical data in the segment module helps the model inherit the temporal feature and GCN helps capture spatial features. The proposed method does not incorporate any other model to capture the temporal feature separately, as using separate models cannot capture the inherent interrelationship between temporal and spatial features. The stacked GCN model can effectively capture temporal and spatial features by employing segmentation.

Table 1.   Day-long (Twenty-four hours) prediction performance for different segments on the SZ-taxi dataset

Number of Segments	RMSE	MAE	R2
24	5.5843	4.4322	0.6995
12	5.6228	4.2262	0.6940
8	5.6179	4.2211	0.6930
6	5.6421	4.2171	0.6876
4	5.6465	4.2279	0.6955
3	5.7155	4.3007	0.6897
2	5.8001	4.3659	0.6808

III.   Experimental Results

A. Dataset Description

In this section, we evaluate the predictive performance of our proposed model using two publicly available real-world datasets: the SZ-taxi dataset and the PeMSD7 dataset. These datasets have gained popularity in traffic forecasting research and have been employed for performance benchmarking in prior studies. Those datasets have both speed and connection data that are needed for GCN.

SZ-taxi: The SZ-taxi dataset, covering taxi trajectories in Shenzhen from January 1 to January 31, 2015, is centered on the Luohu District's 156 highways. This dataset is structured into two essential components: a 156x156 adjacency matrix illustrating highway connections and a feature matrix capturing the time-varying traffic speeds for each road. Each row in the feature matrix corresponds to a unique route, while columns represent traffic speeds at fifteen-minute intervals. The dataset is split into two segments for research purposes, allocating twenty days for training and ten days for testing, facilitating effective model development and evaluation.

PeMSD7: The PeMSD7 dataset provides traffic speed data collected from 228 sensors in California's District Seven during weekdays in May and June 2012. It includes two critical

components: a 228x228 adjacency matrix representing sensor connections within the network and a feature matrix depicting the time-varying traffic speeds for each sensor. Each row in the feature matrix corresponds to an individual sensor, while columns represent five-minute intervals of traffic speed measurements. The dataset is segmented into a training set, consisting of the first month's data encompassing 6,336 timestamps and a test set with an equal number of timestamps, enabling practical model training and evaluation for traffic flow prediction research.

Table 2 illustrates the learning parameters employed in our proposed model. We utilized the Adam optimizer during the training process to minimize the RMSE. The Adam optimizer dynamically adjusts the model's real-time parameters, enhancing its accuracy and computational efficiency. The L2 Regularization technique is used to reduce model overfitting. As we have memory limitations, we used 1000 epochs,64 hidden units, batch size 32, and 0.001 learning rate. As per Table. 1, we can see that twenty-four segments give better performance. Thus, we used twenty-four segments for a one-day prediction.

Table 2. Learning Parameters

Parameter	Description
Learning Rate	0.001
Number of Epoch	1000
Loss Function	RMSE
Hidden Unit	64
Optimizer	Adam
Regularization Techniques	L2 Regularization

B. Evaluation Metrics

To assess prediction performance, we utilize three metrics. Three commonly used performance measurements for model evaluation in various fields are the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). In particular, the RMSE is an important metric to evaluate the effectiveness of the proposed model. The RMSE value indicates the average magnitude of the differences between actual and predicted data values. In general, a smaller RMSE suggests that the model and its predictions perform better, reflecting reduced errors in prediction accuracy. The eqaution of RMSE as in (3).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_1 - \widehat{y_1}^2)} \tag{3}$$

The absolute mathematical operation turns a negative integer into a positive number. Indeed, when calculating the MAE, the absolute difference between an expected (actual) value and a predicted value is always taken, ensuring that the result is positive regardless of whether the prediction overestimates or underestimates the actual value. The formula of MAE as in (4).

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}|y_1 - \widehat{y_1}| \tag{4}$$

The coefficient of determination, often referred to as R-squared, quantifies the proportion of variation in the dependent variable that can be accounted for by the independent variable (s) in a regression model. It has a value between 0 and 1, with higher values suggesting that the model fits the data more closely as in (5).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y_i})^2} \tag{5}$$

C. Compared Methods

We conducted a comparative analysis of our proposed model against several widely recognized models for traffic flow prediction. We selected four commonly employed approaches, encompassing both traditional time-series prediction methods and deep learning techniques.

First is Autoregressive Integrated Moving Average (ARIMA). ARIMA represents a conventional statistical method that captures temporal dependencies within data by employing autoregression,

differencing, and moving average techniques. Researchers have extensively used it for traffic flow estimation [31]. Second is Support Vector Regression (SVR). SVR is a model that forecasts future traffic data by leveraging existing data to train the model and establish the relationship between input and output variables [32]. This model employs a linear kernel function. The next is K-nearest Neighbor (KNN). KNN is a widely recognized supervised learning approach used for data classification based on the proximity of data points to their neighbors [15]. KNN retains all available instances and classifies new cases using a similarity score. The last is Graph Convolutional Network (GCN). GCN represents a semi-supervised deep learning method that captures the spatial characteristics of nodes within a graph. It operates effectively in non-Euclidean spaces, making it suitable for modeling road networks

IV. Result and Discussion

Table 3 shows the performance of the four approaches outlined above and our suggested model on two frequently used datasets. First, we calculate RMSE, MAE, and R2 for a whole day (twenty-four hour) prediction. Table 3 reveals that our proposed approach surpasses the other four methods across both datasets regarding RMSE, MAE, and R2. Lower error values imply higher accuracy, except for R2, where higher values indicate superior performance. The error calculations are conducted twenty-four hours ahead of predictions. In the sz-taxi dataset, our proposed method demonstrates a remarkable 16.9% reduction in RMSE compared to ARIMA and a 9.17% decrease compared to GCN. Moving to the PeMSD7 dataset, our proposed model achieves a substantial 60.4% reduction in RMSE compared to ARIMA, 55.5% reduction compared to SVR, 45.7% reduction to KNN, and 53% reduction to GCN. Our proposed model exhibits superior performance, particularly in the PeMSD7 dataset. This is attributed to the larger size of the PeMSD7 dataset, allowing our model to learn more effectively by relying on historical data for predicting future traffic trends. Notably, ∗ it indicates negligible values, signifying poor prediction performance for the model in those cases.

Table 3. Prediction performance of the proposed model and other baseline models using SZ-taxi data and PeMSD7 datasets for a day (24 hours)

Model Name	SZ-taxi			PeMSD7		
	RMSE	MAE	R²	RMSE	MAE	R²
ARIMA	6.7963	4.6757	*	11.3038	9.1818	*
SVR	6.56454	4.55313	0.6552	10.0653	4.9432	0.5032
KNN	5.96454	4.25313	0.6752	8.2546	4.8241	0.5134
GCN	6.2163	4.6581	0.6451	9.5362	6.8600	0.5162
Proposed Model	**5.6463**	**4.2197**	**0.6871**	**4.4808**	**3.2734**	**0.8105**

The poor results of the baseline methods are because of the difficulty for ARIMA, KNN, and SVR in dealing with complex, irregular time series data. That is why they performed poorly in long datasets like the PeMSD7. Despite utilizing GCN within the model, its predictive performance is subpar. GCN primarily focuses on spatial characteristics, neglecting the temporal nature inherent in traffic data, which is fundamentally time series data. Our proposed model addresses this limitation by segmenting the data, enhancing GCN's ability to handle time series data.

Consequently, our proposed model exhibits superior day-long traffic flow speed prediction capabilities. Additionally, ARIMA, a well-established traffic forecasting method, suffers from reduced prediction accuracy when confronted with extended and irregular data patterns. ARIMA computes its predictions by calculating and averaging errors across individual nodes, and any anomalies in the data can consequently inflate the final total error. On the other hand, in our proposed long-term prediction, error does not propagate, resulting in better results when compared to others.

In Figure 5, we visualize traffic prediction and actual traffic flow for an entire day on one road for the SZ-taxi dataset. The yellow line indicates actual traffic flow, and the blue dotted line indicates predicted traffic flow. The model demonstrates an ability to capture the daily traffic flow data trends. Utilizing GCN for each segmented dataset allows for capturing temporal and spatial characteristics throughout the day.
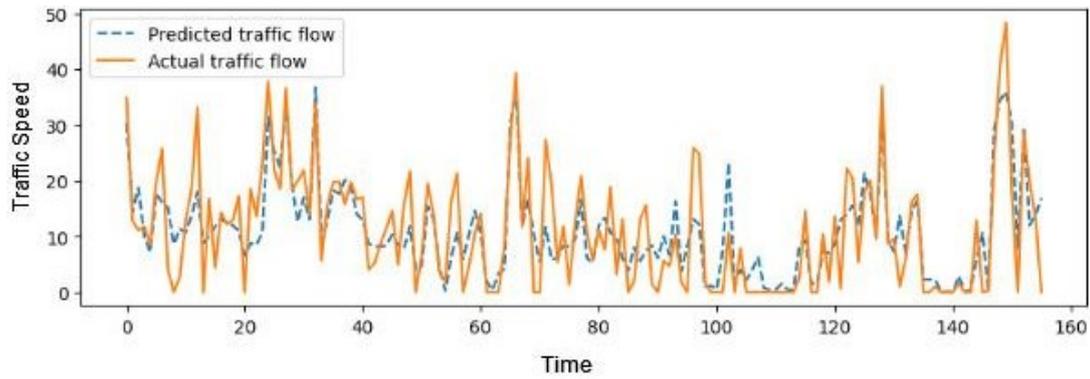
Fig. 5. The visualization results for a prediction horizon of twenty-four hours in the SZ-taxi dataset

Our model has shortcomings as it does not account for external variables such as weather conditions, accidents, or holidays, which can result in limitations in accurately capturing traffic flow dynamics. Our plans involve integrating attention mechanisms to detect abrupt incidents and adopting a dynamic adjacency matrix instead of a static one to enhance the information supplied to the GCN. In addition, we aim to integrate weather conditions and holiday data into our analysis alongside speed data.

V. Conclusion

In this research paper, we introduced the concept of a stacked GCN, a deep learning methodology aimed at tackling the complexities associated with long-term traffic flow prediction. Accurate long-term prediction is essential in traffic management and sustainable urban planning, particularly as urbanization and population growth exacerbate traffic congestion issues. The proposed Stacked GCN model overcomes traditional error accumulation issues by employing a segmented module for temporal feature extraction and leveraging Graph Convolutional Networks' capabilities. Incorporating historical data in segmentation helps our model learn the historical pattern. In a comparison between the ARIMA, SVR, KNN, and GCN models using two real-world traffic datasets, it is evident that the stacked GCN model outperforms the others and yields the most accurate prediction results.

Our model can reduce error from 40% to 60% compared to other methods that we used for comparison. This produces accurate day-long traffic forecasts, providing travelers with preemptive route planning information. Moreover, our model does not use hybrid models like other long-term prediction models, ensuring faster results. In the future, our strategy includes integrating attention mechanisms to detect unexpected events and employing a dynamic adjacency matrix instead of a fixed one to enhance the information available to the GCN. We aim to integrate weather conditions and holiday data into our analysis alongside speed data.

Declarations

Publisher's Note: Department of Electrical Engineering and Informatics - Universitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

[1] M. M. Rahman and N. Nower, "Attention based Deep Hybrid Networks for Traffic Flow Prediction using Google Maps Data," in Proceedings of the 2023 8th International Conference on Machine Learning Technologies, Mar. 2023, pp. 74–81.

[2] M. M. Rahman, A. R. M. Jamil, and N. Nower, "Uncertainty-Aware Traffic Prediction using Attention-based Deep Hybrid Network with Bayesian Inference," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 6, 2023.

[3] D. Rukmana, "Rapid urbanization and the need for sustainable transportation policies in Jakarta," IOP Conf. Ser. Earth Environ. Sci., vol. 124, p. 012017, Mar. 2018.

[4] A. A. Haider, "Traffic jam: The ugly side of Dhaka's development," Dly. Star, vol. 13, 2018.

[5] M. Sweet, "Does Traffic Congestion Slow the Economy?," J. Plan. Lit., vol. 26, no. 4, pp. 391–404, Nov. 2011.

[6] T. Peng, X. Yang, Z. Xu, and Y. Liang, "Constructing an Environmental Friendly Low-Carbon-Emission Intelligent Transportation System Based on Big Data and Machine Learning Methods," Sustainability, vol. 12, no. 19, p. 8118, Oct. 2020.

[7] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting Traffic Congestion Using ARIMA Modeling," in 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Jun. 2019, pp. 1227–1232.

[8] C. P. I. J. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen, "Localized Extended Kalman Filter for Scalable Real-Time Traffic State Estimation," IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1, pp. 385–394, Mar. 2012.

[9] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," Transp. Res. Part C Emerg. Technol., vol. 43, pp. 50–64, Jun. 2014.

[10] Y. Liu and H. Wu, "Prediction of Road Traffic Congestion Based on Random Forest," in 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Dec. 2017, pp. 361–364.

[11] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive Multi-Kernel SVM With Spatial–Temporal Correlation for Short-Term Traffic Flow Prediction," IEEE Trans. Intell. Transp. Syst., vol. 20, no. 6, pp. 2001–2013, Jun. 2019.

[12] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate Multisteps Traffic Flow Prediction Based on SVM," Math. Probl. Eng., vol. 2013, pp. 1–8, 2013.

[13] B. Sharma, V. Kumar Katiyar, and A. Kumar Gupta, "Fuzzy Logic Model for the Prediction of Traffic Volume in Week Days," Int. J. Comput. Appl., vol. 107, no. 17, pp. 1–6, 2014.

[14] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An Improved Bayesian Combination Model for Short-Term Traffic Prediction With Deep Learning," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 1332–1342, Mar. 2020.

[15] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction," Procedia - Soc. Behav. Sci., vol. 96, pp. 653–662, Nov. 2013.

[16] D. Xu, Y. Wang, P. Peng, S. Beilun, Z. Deng, and H. Guo, "Real-time road traffic state prediction based on kernel-KNN," Transp. A Transp. Sci., vol. 16, no. 1, pp. 104–118, Dec. 2020.

[17] K. Kumar, M. Parida, and V. K. Katiyar, "Short Term Traffic Flow Prediction for a Non Urban Highway Using Artificial Neural Network," Procedia - Soc. Behav. Sci., vol. 104, pp. 755–764, Dec. 2013.

[18] A. Koesdwiady, R. Soua, and F. Karray, "Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach," IEEE Trans. Veh. Technol., vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[19] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," pp. 1–14, 2016.

[20] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, "Improved Deep Hybrid Networks for Urban Traffic Flow Prediction Using Trajectory Data," IEEE Access, vol. 6, pp. 31820–31827, 2018.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc., pp. 1–14, 2017.

[22] Z. Chen, B. Zhao, Y. Wang, Z. Duan, and X. Zhao, "Multitask Learning and GCN-Based Taxi Demand Prediction for a Traffic Road Network," Sensors, vol. 20, no. 13, p. 3776, Jul. 2020.

[23] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical Graph Convolution Network for Traffic Forecasting," Proc. AAAI Conf. Artif. Intell., vol. 35, no. 1, pp. 151–159, May 2021.

[24] Y. Xu, Y. Lu, C. Ji, and Q. Zhang, "Adaptive Graph Fusion Convolutional Recurrent Network for Traffic Forecasting," IEEE Internet Things J., no. NeurIPS, pp. 1–12, 2023.

[25] A. Belhadi, Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "A recurrent neural network for urban long-term traffic flow forecasting," Appl. Intell., vol. 50, no. 10, pp. 3252–3265, Oct. 2020.

[26] X. Kong, J. Zhang, X. Wei, W. Xing, and W. Lu, "Adaptive spatial-temporal graph attention networks for traffic flow forecasting," Appl. Intell., vol. 52, no. 4, pp. 4300–4316, Mar. 2022.

[27] Z. Wang, X. Su, and Z. Ding, "Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture," IEEE Trans. Intell. Transp. Syst., vol. 22, no. 10, pp. 6561–6571, Oct. 2021.

[28] Y. Li, S. Chai, Z. Ma, and G. Wang, "A Hybrid Deep Learning Framework for Long-Term Traffic Flow Prediction," IEEE Access, vol. 9, pp. 11264–11271, 2021.

[29] M. Méndez, M. G. Merayo, and M. Núñez, "Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model," Eng. Appl. Artif. Intell., vol. 121, p. 106041, May 2023.

[30] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs Go As Deep As CNNs?," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019, vol. 2019-Octob, pp. 9266–9275, doi: 10.1109/ICCV.2019.00936.

[31] X. Lin and Y. Huang, "Short-Term High-Speed Traffic Flow Prediction Based on ARIMA-GARCH-M Model," Wirel. Pers. Commun., vol. 117, no. 4, pp. 3421–3430, Apr. 2021.

[32] G. Lin, A. Lin, and D. Gu, "Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient," Inf. Sci. (Ny)., vol. 608, pp. 517–531, Aug. 2022.