UG-STNN: A Spatial-Temporal Neural Network Based on Unsupervised Graph Representation Module for Traffic Flow Prediction

Enwei Zhang[1], Zesheng Cheng[1*], Tiankuan Wang[2]and Dongwei Liu[3]

Abstract— **Accurate and efficient traffic flow prediction helps to build an intelligent transportation system and improve the travel experience in daily life. In this study, a new Spatial-Temporal Neural Network Based on Unsupervised Graph Representation Module (UG-STNN) is proposed to improve the graph convolution module, which uses unsupervised learning to extract features in spatial dimensions, and it can learn the structural and feature information in the graph better. Our UG-STNN uses fewer convolutional layers to reduce the number of parameters, decrease the complexity of the model, and improve performance and accuracy. From the experimental results of UG-STNN on different test datasets, the model can approach or even achieve better prediction results compared with other models, which well illustrates the accuracy and stability of the UG-STNN model.**

I. INTRODUCTION

In recent years, with the in-depth integration of cloud computing, Internet of Things, big data, mobile Internet and other electronic information technologies with the transport industry, Intelligent Transportation Systems (ITS) have been gradually constructed and continuously improved in central cities around the world. Accurate and efficient collection of spatial and temporal data related to traffic flow is an important prerequisite for the construction and improvement of ITS. Traffic flow prediction tasks predict the future traffic flow at the target location based on previous traffic patterns and trends, which is then used as the necessary data for subsequent traffic system planning, flow control, accident prevention and control tasks [1]. This process significantly reduces the cost of resources and time required for traffic flow data collection and is therefore one of the main alternatives to traditional methods such as sensors and closed-circuit monitoring.

In the field of traffic flow prediction, the majority of early methods are based on statistical time series models and machine learning models[2-4]. These methods have a limited effect on the extraction of complex spatial-temporal features and are susceptible to noise interference. Consequently, it is challenging to achieve accurate prediction outcomes. In recent years, deep learning methods, in particular those based on spatial-temporal graph neural networks, have yielded significant advances in traffic flow prediction [5]. The graph neural network method employs a graph structure to describe the spatial relationship between traffic network nodes. The combination of temporal neural networks with the extraction of temporal features greatly enhances the accuracy of traffic flow prediction.

Graph Convolutional Network (GCN) [6] is one of the most common modules used in the application of graph neural networks mentioned above in traffic flow prediction tasks. The model can be used to obtain spatial dependencies between nodes by learning how information is transferred between target nodes and their neighbors. By stacking the modules in multiple layers, information is transferred between a target node and its directly connected nodes (one-hop neighbors) first, then indirectly via the one-hop neighboring nodes to the two-hop neighboring nodes, and so on. Since the structure of the model is learned for each target node separately from the information transfer to the relevant nodes in the graph, there is no need for the data to be arranged in a structured manner, which enables more convenient handling of complex unstructured data such as traffic flow. Most of the subsequent models have utilized and improved upon this GCN module, achieving stable and effective prediction performance [7-8].

However, for large-scale and complex traffic networks, due to the complex connectivity relationships presented between nodes, in order to improve the prediction accuracy, a stack of more layers of GCNs is generally required compared to other graph learning tasks. This not only leads to a dramatic increase in the number of parameters in the model, which substantially increases the training cost. At the same time, due to the structure of the GCN itself, the stacking of multiple layers of GCNs tends to smooth the output. According to the statements from the developers, the GCN model may have stable performance before the number of hops reaches 7. After that, with the number of hops increasing, the prediction performance of the GCN-based model will drop dramatically [9]. Fig. 1 below illustrates the results of an experiment where GCN was applied to three public datasets (Cora, Citeseer, and Pubmed) during the study's preparatory phase, which also proves the above statement.

Problems above limits the further improvement of the accuracy of GCN-based traffic flow prediction models. To address these problems, a novel unsupervised graph representation learning module is applied in this study. The module automatically learns the representation of nodes and edges from the complex node connectivity relationships in the graph structure through graph contrastive representation learning, which helps to better extract and apply the implicit features in the graph structure. This effectively avoids the problem of stacking GCN layers due to the complexity of

*Corresponding author : czs_110@hotmail.com
[1]College of Computer Science and Technology, Qingdao University, Qingdao, China
[2]Faculty of Computer Science, University of Alberta, Edmonton, Canada
[3]Menaul School, Qingdao, China

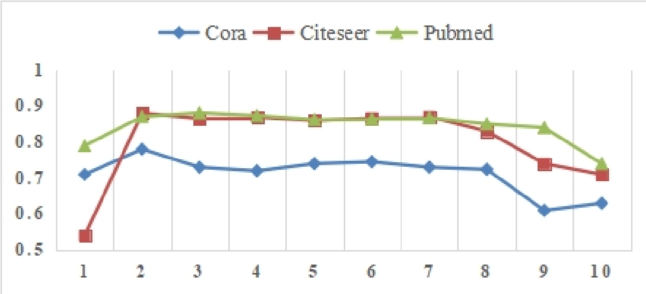

Fig. 1. The effect of different GCN layers

the graph structure, and enables the model to maintain good performance and trainability. Further, this study combines the module with a temporal convolution module to propose the Spatial-Temporal Neural Network Based on Unsupervised Graph Representation Module (UG-STNN) model, and validate it on different datasets in China and abroad.Test results and ablation experiments on the traffic flow prediction task show that the model as well as unsupervised graph representation learning module is stable and effective.

II. RELATED WORK

The traffic flow prediction problem is typically a spatial-temporal data prediction problem, in which the traffic flow at a target location at a target time is not only related to the historical traffic flow at that location (temporal dimension), but also related to the traffic flow at the current moment at its nearby locations (spatial dimension) [10]. Therefore, it is required that the prediction model is able to extract the spatial-temporal information between the data simultaneously and efficiently to achieve better prediction performance.

Compared with the structured time dimension, it is more difficult to extract features in the unstructured spatial dimension, and it has been a hot topic in the research of traffic flow prediction problems in recent years. The introduction of GCN has enabled traffic flow prediction models to make great improvement in prediction accuracy. One of the early representative works that applied GCN to the traffic flow prediction task is the Spatio-Temporal Convolutional Network (STGCN) [11]. The main structure of the model consists of two Temporal Convolutional Network (TCN) [12] modules and a GCN module to extract the spatial and temporal correlations of traffic flow, respectively. Further improvements on this basis include the use of different methods for node neighborhood determination (Diffusion Convolutional Recurrent Neural Network [13] and ST-MetaNet [14]), the use of an adaptive graph representation (Multi-Task Graph Neural Network [15] and Dynamic Graph Convolutional Recurrent Network [16]), and the introduction of Self-attention mechanism (Attention-based Spatial Temporal Graph Convolutional Network [17] and Graph Multi-Attention Network for Traffic Prediction [18]), etc.

As mentioned in Section 1, those GCN-based models do not structurally address the problem of increased model training costs and smoothing of outputs when multiple layers of GCNs are stacked. Unsupervised graph learning [19-23], on the other hand, can use unlabeled data to automatically extract features of the data or data distribution patterns without manual labeling. In contrast to the pre-determination of connectivity between nodes, unsupervised learning methods incorporate a graph learning layer in the model to learn the optimal path for information transfer. This enables information to eventually be passed between nodes that are neighbors at any number of hops according to the pattern learned by the model, regardless of whether they are adjacent or not. Therefore, this study suggests that the use of unsupervised graph learning methods may be an effective means of solving the multilayer GCN stacking challenge.

However, the above representative works mainly focus on link representation learning of graphs (edge level) and overall representation learning of graphs (graph level). It is not applicable to the node-level task of traffic flow prediction. Therefore, this study proposes a node-level, graph representation learning method based on generating different views for comparison. The aim is to more fully extract and utilize the potential information of the graph structure to enhance the efficiency and accuracy of subsequent prediction tasks.

III. METHODOLOGY

This section describes the structure of Unsupervised Graph Representation Module, Temporal Convolutional Module and overall structure of UG-STNN model.

A. Unsupervised Graph Representation Module

The spatial convolution module of UG-STNN is implemented by unsupervised learning at the node level. The core process of this module is to learn by comparing the structural similarity of two new graphs generated based on the original graph. Specifically, for the input graph structure, the original graph is first randomly destroyed to generate two subgraphs. As illustrated in Fig. 2, the destruction is done by removing some edges and masking some features of all nodes in the same generated subgraphs.

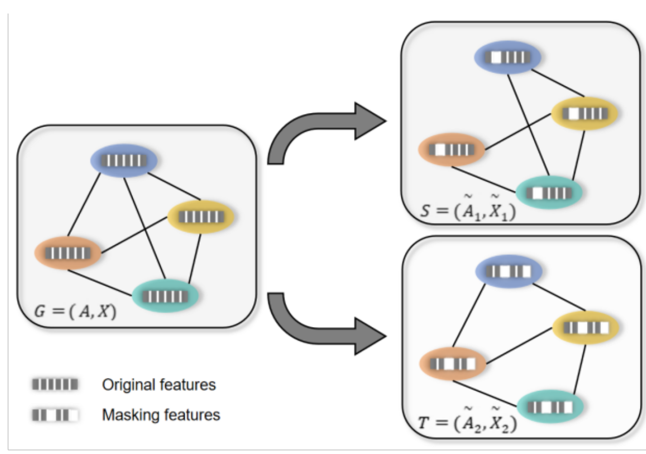

Fig. 2. Removing edges and masking features

For edge removal, a matrix $M \in \{0,1\}^{N\times N}$is randomly generated, whose elements obey the Bernoulli distribution $M_{ij} \sim \mathrm{B}(1,p_l)$, where p_l is the probability that each edge is removed. Let A be the adjacency matrix of the original graph. Then the Hadamard product of A and M, i.e. $\widetilde{A} = A \circ M$, is the adjacent matrix of the generated subgraph.

Similarly, for the masking of node features, the matrix $m \in \{0,1\}^F$is randomly generated, whose elements obey the Bernoulli distribution $m_i \sim \mathrm{B}(1,p_n)$, where p_n is the probability that each feature is masked. Let X be the feature of node in the original graph. Then the Hadamard product of X and m, i.e. $\widetilde{X} = X \circ m$ is the node features of the generated subgraph.

After generating the two subgraphs, the next step is to compute the embedding, $S = \varphi(\widetilde{A_1},\widetilde{X_1})$and $T = \varphi(\widetilde{A_2},\widetilde{X_2})$of the two subgraphs to represent the nodes in the graph. There are different available choices of encoder φ. Based on the properties of traffic network, GCN is used as a trainable encoder in UG-STNN.

Next, the generated two subgraphs are compared for node consistency. As shown in Fig. 3 below, same nodes (s_i and t_i) in the two subgraphs are defined as positive node pairs, while different nodes (s_i and other nodes) in two subgraphs are defined as negative node pairs.

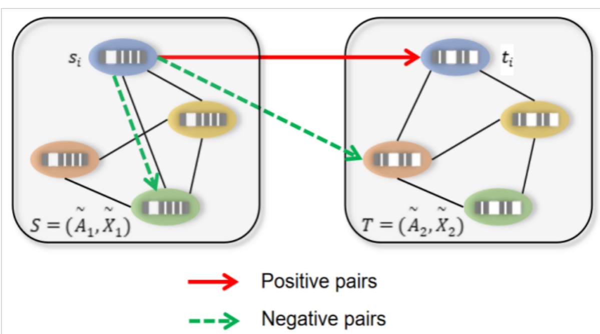

Fig. 3. Comparison of nodes

Accordingly, the consistency of a positive node pair can be defined as:

$$\phi(s_i,t_i) = \log \frac{e^{\frac{\rho(s_i,t_i)}{\eta}}}{e^{\frac{\rho(s_i,t_i)}{\eta}} + \sum_{n=1}^{N} \delta_{[n\neq i]} e^{\frac{\rho(s_i,t_n)}{\eta}}} \tag{1}$$

$$\rho(s,t) = c(f(s),f(t)) \tag{2}$$

where:

f : Trainable multilayer perceptron function (MLP)

c : Cosine similarity function

η: Adjustable temperature hyper-parameter

δ: Binary function, its value equal to 0 when $n = i$, equal to 1 otherwise.

Φ: Consistency of the node pair.

Based on the above definition of node consistency, the proposed module can learn the optimal graph representation by maximizing the similarity of positive samples and minimizing the similarity of negative samples. The parameters of the trainable GCN and MLP are adjusted to make similar node pairs closer together in the representation space, while dissimilar node pairs are more scattered. The objective function used in this model is to maximize the average of the node consistency of positive node pairs and is deformed as shown in Eq. (3) to facilitate the gradient calculation.

$$\max \omega = \frac{1}{2N}\sum_{i=1}^{N}[\phi(s_i,t_i)+\phi(t_i,s_i)] \tag{3}$$

Overall, the algorithm of the proposed Unsupervised Graph Representation Module is: Firstly, two subgraphs are generated by destroying the original graph. Then the nodeembedding is performed by applying GCN. Based on the embedding results, the consistency of positive node pairs can be calculated as the objective function to train the parameters of GCN and MLP.

B. Temporal Convolution Module

The UG-STNN uses a temporal convolution module as shown in Fig. 4 below to extract temporal information. The module consists of a two-dimensional temporal convolution layer to extract short-time temporal correlations, and a Dilated convolution layer with a large sensory field to capture long-range dependencies.

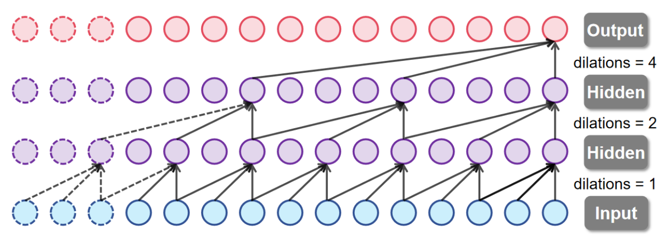

Fig. 4. Temporal Convolution Module

C. Structure of UG-STNN

The overall structure of the UG-STNN model is shown in Fig.5. The unsupervised learning module is first used to extract the graph structure features, and then the temporal dimension is convolved. The spatial convolution can capture the spatial relationship between nodes and extract the spatial feature representation of nodes at target time slot. Temporal Convolution allows further processing of these spatial features in the time dimension, capturing the dynamically changing features of the nodes between different time intervals. The model also contains an output layer (nonlinear activation function, normalization, fully connected) to process the results after the spatial and temporal modules.

IV. EXPERIMENT AND ANALYSIS

The main contents of this section include experimental preparation, experimental results and analyses.

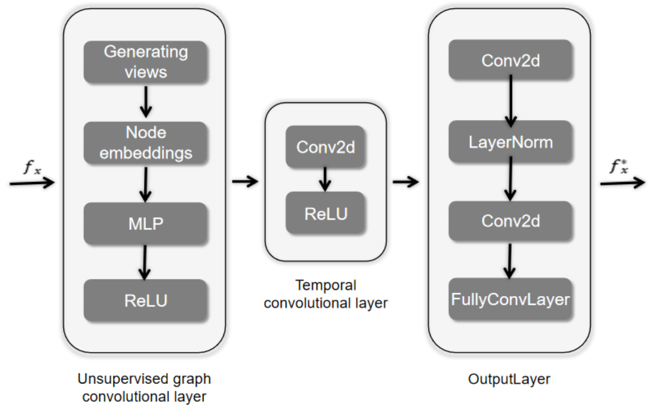

Fig. 5. Structure of UG-STNN

A. Experimental Preparation

1) Problem Definition: The traffic flow prediction task can be defined as: give the historical measurements of all the nodes on the traffic network over past t time slices to predict future traffic flow sequences $F = (f^1, f^2, \ldots, f^N) \in R^{N*p}$of all the nodes on the whole traffic network over the next p time slices, where $f^i = (f^i_{t+1}, f^i_{t+2}, \ldots, f^i_{t+p}) \in R^p$ denotes the future traffic flow of node i from $t + 1$.

2) Experimental Environment: All the experiments in this study are conducted on a server with a single Intel Xeon W-2133@3.6Hz CPU and one 32 RAM NVIDIA V100 GPU.

3) Dataset: Two public datasets shown in the following TABLE I are applied to the experiments in this section. The dataset is divided into three distinct subsets: a training set, a validation set, and a test set, allocated in a proportional distribution of 7:1:2. Since it involves time series data analysis, the division is done by taking successive values according to the time series. The predictive model undertakes the extraction of pertinent information from the preceding 3, 6 and 12 temporal intervals, consequently forecasting forthcoming traffic flow dynamics at the 3rd, 6th, and 12th intervals (15min, 30min and 60min), respectively. Employing the identical dataset, pertinent models are trained and assessed, with the final evaluation derived from the average performance across 10 iterations of testing.

4) Baseline: The following six models: STGCN, ASTGCN, GMAN, DCRNN, MTGNN and DGCRN are used as baseline models in this study. Their underlying theory associated with these models has been briefly described in the related work section. Except for the necessary modifications to the hyper-parameters such as the dimensions of input and output that ensure program operation. The rest of the hyper-parameters such as network composition, learning rate, masks, etc. use the optimal values provided in the open source code of the methods.

5) Evaluation Indicators: Following three evaluation indicators are used in this study to assess the performance of the model:

Root Mean Square:

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{n}\left(h(x_i) - y_i\right)^2}$$

Mean Absolute Error:

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{n}|h(x_i) - y_i|$$

Mean Absolute Percentage Error:

$$\text{MAPE} = \frac{100}{m}\sum_{i=1}^{n}\left|\frac{h(x_i) - y_i}{y_i}\right|$$

B. Experiment Results

TABLE II presents the results of the testing of the six baseline models and the proposed UG-STNN model. In addition, the UG-STNN-A model is used as the ablation experimental model and its results are also presented in TABLE II. The model uses linear layers to replace the graph representation learning module in the original UG-STNN model to validate the effectiveness of the module.

Fig. 6 compared the prediction results of different models in the same dataset (METR-LA), at the same time interval (15 min) for a specific vertex. In these six subfigures, the horizontal axis represents time intervals and the vertical axis represents traffic flow. The red line in the figure represents the ground truth value. In other words, the closer the predicted value is to the red line, the better the prediction performance of the model.

Fig. 7 presents a comparison of the prediction results of different models in the same dataset (METR-LA), at the same time slice for all vertices. For clarity, only MTGNN and DGCRN, the top two ranked baseline model are presented. The ground truth is distributed on the line of $y = x$. That is, the more convergent the results to the line $y = x$, the better the prediction performance is, and vice versa.

C. Analysis

As it can be seen in TABLE II, UG-STNN achieves the state-of-the-art results on more than half of the total nine test sets (three datasets * three time slices) and evaluation matrices, with some of the results even outperforming baseline model by more than 10%. UG-STNN also ranked in the top three on the remaining test sets. Therefore, the effectiveness of UG-STNN and the proposed Unsupervised Graph Representation Module, in traffic flow prediction tasks can be well illustrated.

Fig. 6 shows that UG-STNN can extract features and make reasonable predictions regardless of whether the traffic flow is in the peak, valley or normal range. However, the longer the time span of the prediction (from 3 slices to 6 slices then to 12 slices), the smoother the prediction results become. At this point, the model becomes less sensitive to small fluctuations and its predictions become less accurate. This shortcoming is inherent to the time module itself. Solving the long-range dependence problem in time series analysis has proven to be challenging.

Fig. 7 shows that compared with the baseline models, the advantage of UG-STNN is that the prediction results are more accurate when the flow of the target vertex is high or low. The reason for the worse prediction results

TABLE I
DATASETS

Dataset	No. of Nodes	No. of Time Slices	Description
PEMSD7M [11]	228	12672	District 7, the state of California
METR-LA [24]	207	34272	Los Angeles Metropolitan
PEMSBAY [25]	325	52116	Bay Area, the state of California

TABLE II
EXPERIMENT RESULT

Datasets	**Model**	**Time Steps**								
		3 Steps (15 Min)			**6 Steps (30 Min)**			**12 Steps (60 Min)**		
		MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
PEMS7M	STGCN	5.364	3.496	6.40%	6.698	4.243	8.21%	7.863	4.795	9.96%
	ASTGCN	5.457	3.57	6.95%	6.638	3.981	8.75%	7.745	4.676	10.73%
	GMAN	6.625	3.947	8.35%	7.245	4.243	8.98%	8.834	5.017	11.57%
	DCRNN	5.733	4.546	6.68%	6.867	3.965	8.54%	8.387	4.703	10.92%
	MTGNN	5.232	3.27	6.13%	6.453	3.834	7.87%	7.641	**4.354**	9.44%
	DGCRN	5.172	**3.134**	6.24%	**6.034**	**3.765**	7.83%	7.845	4.397	9.63%
	UG-STNN	**5.073**	3.301	**6.07%**	6.546	3.782	**7.33%**	**7.25**	4.369	**9.37%**
	UG-STNN-A	7.262	4.859	10.01%	9.788	6.351	12.85%	12.974	6.420	13.77%
METR-LA	STGCN	7.402	4.697	10.31%	9.453	5.356	11.78%	10.684	5.675	13.61%
	ASTGCN	7.398	4.735	9.78%	8.622	5.413	11.38%	11.004	5.961	14.56%
	GMAN	9.785	6.786	11.05%	10.673	6.941	11.93%	11.768	8.638	13.83%
	DCRNN	7.292	4.645	9.86%	8.56	5.347	11.66%	10.453	5.945	13.43%
	MTGNN	7.253	4.296	10.12%	8.466	4.503	11.67%	10.127	5.387	13.14%
	DGCRN	7.248	**4.267**	**9.05%**	8.35	**4.359**	10.85%	9.837	**5.241**	13.26%
	UG-STNN	**7.083**	4.271	9.18%	**8.149**	4.393	**10.32%**	**9.345**	5.253	**12.96%**
	UG-STNN-A	9.339	6.328	12.55%	11.062	7.011	14.75%	14.825	8.833	15.03%
PEMSBAY	STGCN	3.533	1.899	3.54%	4.492	2.349	4.95%	5.998	3.231	5.96%
	ASTGCN	3.347	1.723	3.47%	4.402	2.269	4.89%	5.574	2.847	5.47%
	GMAN	3.301	1.876	4.65%	4.037	2.256	5.067%	4.985	2.632	5.78%
	DCRNN	3.078	1.764	3.27%	3.956	2.187	4.55%	5.416	2.764	5.95%
	MTGNN	2.967	1.634	3.06%	3.750	1.923	4.18%	5.023	2.541	5.34%
	DGCRN	2.753	**1.569**	**2.87%**	3.738	1.894	4.26%	4.987	**2.394**	5.27%
	UG-STNN	**2.740**	1.684	2.93%	**3.693**	**1.712**	**4.01%**	**4.816**	2.467	**5.04%**
	UG-STNN-A	3.601	1.958	3.57%	4.487	2.251	4.87%	5.894	3.194	5.86%

Fig. 6. Comparison between different model (Same dataset and vertex)

of the baseline model is the appearance of smoothness in the model output caused by the stacking of multiple layers of GCN modules. In terms of spectrogram theory, GCN filters out the high-frequency components (i.e., data with large feature differences) from the input data and retains the low-frequency portion (data with small feature differences). And therefore multiple layers of GCN stacking ultimately result in smoothing and homogenization of the output data. The results of this experiment are further evidence that the introduction of unsupervised graph representation is an effective means to solve this problem.

To sum up, it is believed that the proposed model UG-STNN is an elegant and effective supplements to graph-based deep learning model for traffic flow prediction tasks.

V. CONCLUSIONS

In summary, this study argues that current traffic flow prediction models based on graph neural networks suffer from high training costs and smoothed outputs due to the multi-layer stacking of GCN models. To address this issue, a new node-level unsupervised graph representation module has been designed to extract the spatial correlation of traffic flow data. Combined with the spatial convolution module, a

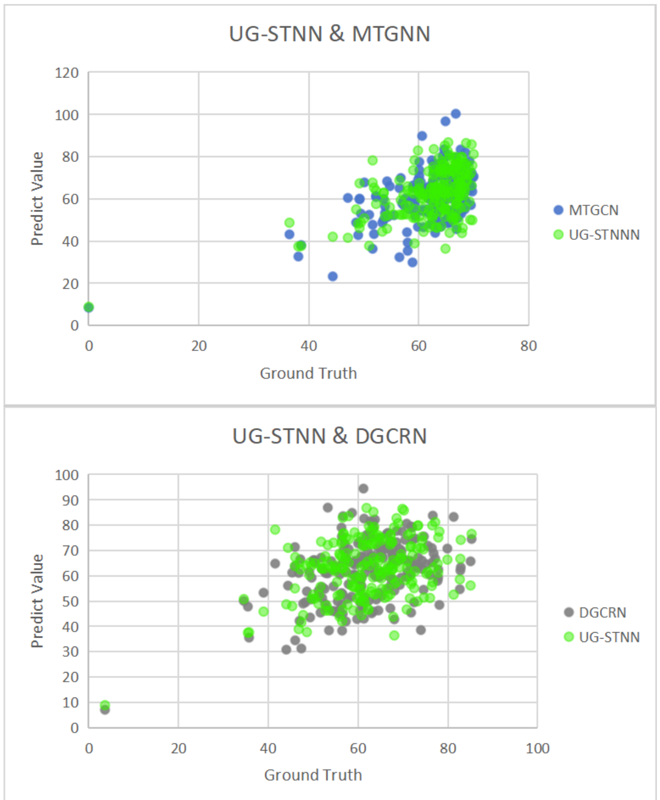

Fig. 7. Comparison between UG-STNN & DGCRN (Same dataset and time slice)

new UG-STNN model has been proposed for traffic flow prediction tasks. Experiments have shown that the model performs well on different test datasets, which demonstrating its efficiency and stability.

In addition, this study has limitations as the datasets used are limited to traffic flow data from California and the Los Angeles metropolitan area. This type of data is typically more stable, which may affect the model's ability to handle factors that can impact traffic flow, such as unexpected events and anomalies. Future research could focus on enhancing the model's spatial modeling and generalization abilities, improving its interpretability, and refining its ability to handle abnormal situations. These improvements would further enhance the model's predictive performance and practical application value.

ACKNOWLEDGMENT

This work was supported in part by The Natural Science Foundation of Shandong Province Grant No. ZR2022QF144

REFERENCES

[1] Lv, M., et al., Temporal multi-graph convolutional network for traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 2020. 22(6): p. 3337-3348.
[2] Khan, N.U., et al., Traffic flow prediction: an intelligent scheme for forecasting traffic flow using air pollution data in smart cities with bagging ensemble. Sustainability, 2022. 14(7): p. 4164.
[3] Liu Y, Wu F, Liu Z, et al. Can language models be used for real-world urban-delivery route optimization?. The Innovation, 2023, 4(6).
[4] Mohammadian S, Zheng Z, Haque M M, et al. Continuum modeling of freeway traffic flows: State-of-the-art, challenges and future directions in the era of connected and automated vehicles. Communications in Transportation Research, 2023, 3: 100107.
[5] Xu M, Di Y, Ding H, et al. AGNP: Network-wide short-term probabilistic traffic speed prediction and imputation. Communications in Transportation Research, 2023, 3: 100099.
[6] Bhatti, T., et al. Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence. International Journal of Intelligent Systems, 2023, 2023(1): 8342104.
[7] Zhou, J., et al., Graph neural networks: A review of methods and applications. AI open, 2020. 1: p. 57-81.
[8] Xing, Y., et al. Learning hierarchical graph neural networks for image clustering. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[9] Lv Y, Lv Z, Cheng Z, et al. TS-STNN: Spatial-temporal neural network based on tree structure for traffic flow prediction. Transportation research part E: logistics and transportation review, 2023, 177: 103251.
[10] Lu, S., et al., A combined method for short-term traffic flow prediction based on recurrent neural network. Alexandria Engineering Journal, 2021. 60(1): p. 87-94.
[11] Yu, B., H. Yin, and Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875, 2017.
[12] Lea, C., et al. Temporal convolutional networks: A unified approach to action segmentation. in Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14. 2016. Springer.
[13] Li, Y., et al., Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, 2017.
[14] Pan, Z., et al. Urban traffic prediction from spatio-temporal data using deep meta learning. in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.
[15] Pasini, M.L., et al., Multi-task graph neural networks for simultaneous prediction of global and atomic properties in ferromagnetic systems. Machine Learning: Science and Technology, 2022. 3(2): p. 025007.
[16] Li, F., et al., Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. ACM Transactions on Knowledge Discovery from Data, 2023. 17(1): p. 1-21.
[17] Guo, S., et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. in Proceedings of the AAAI conference on artificial intelligence. 2019.
[18] J. Zheng, C., et al. Gman: A graph multi-attention network for traffic prediction. in Proceedings of the AAAI conference on artificial intelligence. 2020.
[19] Mo, P., et al. Simple unsupervised graph representation learning. Proceedings of the AAAI conference on artificial intelligence. 2022, 36(7): 7797-7805.
[20] Zhou, H., et al. Multiview deep graph infomax to achieve unsupervised graph embedding. IEEE Transactions on Cybernetics, 2022, 53(10): 6329-6339.
[21] Zhao, Y., et al. Unsupervised structure-adaptive graph contrastive learning. IEEE Transactions on Neural Networks and Learning Systems, 2023.
[22] Zhou, W., et al. Unsupervised Discriminative Feature Selection via Contrastive Graph Learning. IEEE Transactions on Image Processing, 2024.
[23] Fang, H., et al. Joint multi-view unsupervised feature selection and graph learning. IEEE Transactions on Emerging Topics in Computational Intelligence, 2023.
[24] Wu Z, Pan S, Long G, et al. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121, 2019.
[25] Jiang, R., et al. Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction. in Proceedings of the 30th ACM international conference on information & knowledge management. 2021.