

TrafficStream: A Streaming Traffic Flow Forecasting Framework Based on Graph Neural Networks and Continual Learning

Xu Chen*, Junshan Wang, Kunqing Xie

School of Electronics Engineering and Computer Science, Peking University, Beijing, China
 {sylover,wangjunshan,kunqing}@pku.edu.cn

Abstract

With the rapid growth of traffic sensors deployed, a massive amount of traffic flow data are collected, revealing the long-term evolution of traffic flows and the gradual expansion of traffic networks. How to accurately forecasting these traffic flow attracts the attention of researchers as it is of great significance for improving the efficiency of transportation systems. However, existing methods mainly focus on the spatial-temporal correlation of static networks, leaving the problem of efficiently learning models on networks with expansion and evolving patterns less studied. To tackle this problem, we propose a *Streaming Traffic Flow Forecasting Framework*, **TrafficStream**, based on *Graph Neural Networks (GNNs)* and *Continual Learning (CL)*, achieving accurate predictions and high efficiency. Firstly, we design a traffic pattern fusion method, cleverly integrating the new patterns that emerged during the long-term period into the model. A JS-divergence-based algorithm is proposed to mine new traffic patterns. Secondly, we introduce CL to consolidate the knowledge learned previously and transfer them to the current model. Specifically, we adopt two strategies: historical data replay and parameter smoothing. We construct a streaming traffic dataset to verify the efficiency and effectiveness of our model. Extensive experiments demonstrate its excellent potential to extract traffic patterns with high efficiency on long-term streaming network scene. The source code is available at <https://github.com/AprLie/TrafficStream>.

1 Introduction

In recent years, the development of the economy and society has prompted the construction of transportation facilities like roads, toll stations, subway stations, etc. To monitor the real-time traffic status of these facilities, the number of traffic sensors deployed also gains rapid growth. As a result, more high-quality traffic flow data is recorded, revealing the expansion of traffic networks and the evolution of traffic flows.

Traffic flow forecasting is one of the most crucial tasks of Intelligent Transportation System (ITS), which significantly affect our daily routines. Nevertheless, the intricate spatial-temporal correlation of traffic flow raises the difficulty of accurate forecasting. To tackle this problem, researchers introduce various cutting-edge methods into this field, such as CNNs [Zhang *et al.*, 2017], LSTMs [Li *et al.*, 2017] and Transformers [Wang *et al.*, 2020c]. Impressed by the promising performance of GNNs on graph structure data, researchers are devoted to integrating GNNs into traffic flow forecasting model for mining dependency on spatial-temporal aspect, which becomes a great success [Yu *et al.*, 2017; Song *et al.*, 2020; Chen *et al.*, 2020]. However, along with the improvement of model performance, there comes a considerable increase in the number of parameters. Besides, they still face the challenges of long-term streaming traffic networks.

The long-term streaming networks, referring to traffic networks with changing flows of nodes and expanding structures in a long-term period, i.e., several years, have been less studied, whereas the real-world scene is like this. The simplest way is to retrain a model every year. However, the massive amount of parameters of a new model are severe challenges to computing resources and the amount of data, especially when datacenters store data in a queue-like manner. Besides, it will be difficult to generate accurate predictions via less data from new added stations. Considering that previous models have been well trained and for resource conservation, it is natural to transfer their knowledge to the model in the current year.

Directly using previous models still needs to address two critical problems. Firstly, previous models should accommodate the new data. Unlike short-term data whose trends are more stable and regular, traffic flow patterns gradually evolve in the long-term streaming networks. The inconsistency between patterns of new data and historical data will reduce the accuracy of forecasting. Besides, due to network expansion, the topological structures of traffic networks corresponding to the new data are also different from the previous ones. So the spatial dependency captured by previous models may not be suitable anymore. Secondly, knowledge extracted by previous models should be consolidated as there is rich information about previously learned patterns¹. Therefore, *how to*

¹Previously learned patterns include long-term stable patterns and temporarily disappearing patterns.

*Corresponding Author

efficiently capture patterns from new data while consolidating historical knowledge becomes our motivation.

For efficiency and effectiveness, we propose a Streaming Traffic Flow Forecasting Framework based on GNNs and Continual Learning, TrafficStream. On the one hand, we design a traffic pattern fusion approach to integrate the unknown patterns of new nodes and changing patterns of existing nodes into the model. To discover these new patterns, we propose an algorithm based on JS-divergence to measure the changes of node features depending on the model. On the other hand, in order to consolidate historical traffic knowledge and transfer it to the current model for better prediction, we adopt strategies of historical information replay and parameter smoothing from data and model perspectives, respectively. Finally, a long-term streaming network flow dataset is constructed to verify the performance of TrafficStream.

Our main contributions are summarized as follows:

- We explore the problem of streaming traffic flow forecasting based on GNNs. To our best knowledge, it is the first time that GNNs are introduced in the scenario of changing flows of nodes and expanding network structures in long-term streaming traffic networks.
- We propose a novel unified framework for GNN-based models to address the difficulty of fusing historical knowledge and new flow data. We design different components for training models on expanding networks and nodes with new evolutionary patterns. Information replay and knowledge smoothing components are adopted to consolidate historical knowledge.
- To verify the effectiveness and efficiency of our framework, we construct a long-term streaming network dataset, PEMS3-Stream, based on real-world traffic data. The performance indicates the superiority of our model in consuming time with comparative accuracy.

2 Related Work

2.1 Traffic Flow Forecasting

Recent years have witnessed the rapid development of ITS, attracting researchers to solve one of the key problems, traffic flow forecasting. Statistical models, such as VAR [Zivot and Wang, 2006] and SVR [Castro-Neto *et al.*, 2009] simplify the forecasting into individual time-series prediction. These methods failed to capture the spatial-temporal correlation between different locations. In addition, training such a model for all locations results in poor performance, while each location with a model will increase cost in both consuming time and storage. Along with the development of deep learning, researchers apply various frameworks to overcome the difficulty of understanding the pattern of spatial-temporal data. DMVST-Net [Yao *et al.*, 2018] integrates CNN and long-short term memory (LSTM) to extract spatial and temporal dependencies. STGCN [Yu *et al.*, 2017] introduces graph convolution for mining spatial patterns and CNN with the gated mechanism for temporal features. Graph WaveNet [Wu *et al.*, 2019] is proposed to adaptively learn the spatial adjacency matrix and expand the receptive field along the time axis. STSGCN [Song *et al.*, 2020] simultaneously models the

spatial-temporal correlation via a localized spatial-temporal graph. Although these methods achieve impressive results on traffic flow forecasting task, they are not designed for long term streaming network scene.

2.2 Continual Learning on Graphs

Continual learning, also called as life-long learning and incremental learning, is a technique to train the model sequentially when data from different tasks arrives in a streaming manner. Generally, existing methods can be divided into three categories: regularization-based methods [Kirkpatrick *et al.*, 2017], replay-based methods [Rolnick *et al.*, 2019] and parameter isolation methods [Rusu *et al.*, 2016], and all of them aim to learn knowledge from new tasks without forgetting knowledge from previously trained tasks. Traditional continual learning methods mainly focus on image processing, assuming that samples are independent to each other.

Recently, continual learning has been applied in the field of graph learning, which can deal with the efficient update of graph models, especially on large-scale dynamic graphs. ContinualGNN [Wang *et al.*, 2020b] proposes an incremental GNN model on streaming graphs by detecting new coming patterns efficiently and consolidating existing patterns via continual learning. Besides, ER-GNN [Zhou *et al.*, 2020] and Feature Graph [Wang *et al.*, 2020a] presents different experience replay strategies for continual graph representation learning. However, these methods only consider the incremental learning of GNN models when the graph structures change, and thus can only be applied to scenarios such as social networks and knowledge graphs. They fail to handle the effective update of the spatial-temporal model in traffic networks when the temporal patterns of stations and the topological structures of networks evolve simultaneously.

3 Preliminaries

The long term streaming traffic network is represented as $G = (G_1, G_2, \dots, G_{\mathcal{T}})^2$, where $G_{\tau} = G_{\tau-1} + \Delta G_{\tau}$ for all time intervals τ , meaning the network keeps expanding over time. Traffic network of the τ -th year can be denoted as a graph $G_{\tau} = (V_{\tau}, E_{\tau}, A_{\tau})$, where V_{τ} is a finite node set with $|V_{\tau}| = N_{\tau}$ nodes. Nodes on the graph can represent sensors or stations. E_{τ} represents the edge set with edges connecting nodes from V_{τ} and $A_{\tau} \in \mathbb{R}^{N_{\tau} \times N_{\tau}}$ is the adjacency matrix. Devices deployed at nodes will periodically measure and record traffic status features, including flows, speed and occupancy. Flow data of all nodes on graph G_{τ} observed at time-step t can be denote as $X_{\tau}^t \in \mathbb{R}^{N_{\tau} \times F}$, where F is the number of traffic status features.

The purpose of traffic flow forecasting is to provide accurate predictions of future traffic flow when historical data is given, which can be formulated as:

Problem 1. Given the long-term streaming traffic network $G = (G_1, G_2, \dots, G_{\mathcal{T}})$ with $G_{\tau} = (V_{\tau}, E_{\tau}, A_{\tau})$ and the corresponding historical data $X_{\tau} = (X_{\tau}^1, \dots, X_{\tau}^T)$, we are

²In this paper, we use $\tau \in \{1, 2, \dots, \mathcal{T}\}$ to represent a long time interval that the traffic network structure remains unchanged within τ . $t \in \{1, 2, \dots, T\}$ denotes the time-steps traffic data is observed.

expected to learn a series of function $\Psi = (\Psi_1, \Psi_2, \dots, \Psi_T)$ for prediction of traffic flow series $\hat{Y}_\tau = \Psi_\tau(X_\tau) = (\hat{Y}_\tau^{T+1}, \dots, \hat{Y}_\tau^{T+K})$ for all N_τ nodes in the next K time-steps after time-step T , i.e.,:

$$\Psi_i^* = \arg \min_{\Psi_i} \|\Psi_\tau(X_\tau) - Y_\tau\|^2, \quad (1)$$

where $Y_\tau^T \in \mathbb{R}^{N_\tau}$ denotes the ground-truth of the traffic flow at time-step T and $Y_\tau = (Y_\tau^{T+1}, \dots, Y_\tau^{T+K})$ represents the time series to be forecast.

4 TrafficStream

In this section, we elaborate on the proposed framework of streaming traffic flow forecasting based on *GNNs* and *Continual Learning*, **TrafficStream**. We firstly introduce a simple GNN model as a surrogate model of those complex traffic flow forecasting methods. Our framework balances between learning new traffic patterns from new data and maintaining the historical knowledge learnt from previous data. New nodes with its 2-hops neighbors are used to construct a sub-graph for mining the influence of network expansion. Existing nodes whose traffic patterns change significantly are detected through an algorithm based on JS-divergence. To consolidate the previous knowledge, historical nodes of traffic networks are replayed and weighted smoothing constraints are imposed on the current training model. The details of TrafficStream are described in the following.

4.1 Surrogate Traffic Flow Forecasting Model

The great success of graph neural networks (GNNs) have attracted the attention of researchers and they are introduced to capture spatial and temporal features of traffic flow data. Consequently, current studies on traffic flow forecasting are mainly based on GNNs: they are leveraged to extract spatial dependency between nodes together with sequence models like CNNs and RNNs mining temporal trends of traffic flow [Yu *et al.*, 2017; Wu *et al.*, 2019; Guo *et al.*, 2019]. Recent studies simultaneously capture spatial-temporal correlation through applying GNNs on spatial-temporal fusion graphs constructed by data [Song *et al.*, 2020; Mengzhang and Zhanxing, 2020].

To intuitively describe our framework, we follow their common design and propose a simple surrogate traffic flow forecasting model (**SurModel**) to represent those complex ones deployed in ITS. SurModel is composed of two GNN layers with a CNN layer sandwiched in between. Denote the input of l -th GNN layer in τ -th year as $H_\tau^l \in \mathbb{R}^{N_\tau \times C_\tau^l}$ ($H_i^0 = X_\tau$), spatial information obtained by the GNN layer can be formulated as:

$$H_\tau^{l+1} = \sigma(A_\tau H_\tau^l W_{\tau 1}^l + I H_\tau^l W_{\tau 2}^l) \quad (2)$$

where $W_{\tau 1}^l, W_{\tau 2}^l \in \mathbb{R}^{C_\tau^l \times C_\tau^{l+1}}$ are learnable parameters. $I \in \mathbb{R}^{N_\tau \times N_\tau}$ is identity matrix and σ is the activation function. 1D CNN is adopted to capture temporal patterns on the output of the first GNN layer and generate features for the second GNN layer. We apply skip connection to concatenate input X_τ with output of the second GNN layer H_τ^2 . The result will be fed into a fully connected layer to generate traffic flow in

the next K time-steps. SurModel is set to minimize the L2-loss between predicted value \hat{Y}_τ and ground-true value Y_τ :

$$\mathcal{L}(\hat{Y}_\tau, Y_\tau) = \|\hat{Y}_\tau - Y_\tau\|^2. \quad (3)$$

4.2 New Traffic Pattern Fusion

Traffic flow patterns are believed to be stable without accidents on the roads. Researchers tend to categorize them into daily pattern, weekly pattern, monthly pattern and so on. Based on this assumption, they are devoted to decomposing traffic flow into different patterns for deeper comprehension of traffic systems and better prediction of traffic status. However, this assumption becomes invalid if we observe the flow in a long term: trends may evolves sluggishly over time and it is also possible that the network structures change due to deployment of new sensors or stations. The extracted patterns are not more accurate enough to describe traffic laws of the current network. To efficiently update these patterns under such situations, we propose two components for training on expanding networks and better extracting evolving patterns.

Expanding Network Training

Current GNN based traffic flow forecasting models have a huge amount of parameters, and the time complexities are $\mathcal{O}(N_{\tau-1}^2)$ which can be attributed to matrix multiplication of GNN layers in Eq. (2). When network scale changes, retraining the model definitely raises the complexity to $\mathcal{O}(N_\tau^2) = \mathcal{O}((N_{\tau-1} + \Delta N_\tau)^2)$. Here ΔN_τ is the node set that only contains newly added nodes. In real-world traffic network, it is often the case that previous networks are sufficiently large while the newly deployed nodes only account for a small part. Thus, retraining seems more time consuming as $\Delta N_\tau \ll N_\tau$. Motivated by the scale between ΔN_τ and N_τ as well as online learning methods, we firstly use ΔN_τ to update the trained SurModel $\Psi_{\tau-1}$ as Ψ_τ . Note that GNN layers aim to capture the dependency between nodes and its neighborhoods, we generate a sub-graph by selecting neighbor nodes of newly added nodes on G_τ within 2-hops. The complexity is reduced to $\mathcal{O}((\Delta N_\tau d^2)^2)$ where d is the average degree of nodes. Remember that on a traffic network nodes are connected to 2 nodes on a straight road and 4 at a crossroads, so that d is around 2~4 and $\mathcal{O}((\Delta N_\tau d^2)^2)$ is much smaller than $\mathcal{O}(N_\tau^2)$.

Evolution Pattern Detection

In addition to the effect of network expansion, traffic patterns would not remain constant: people may change residences and workplaces or there may be some changes at the point of interest (POI), and thus travelling plans vary over time. It is necessary to detect nodes where current patterns sharply conflict with previous ones. We design a node detection algorithm to discover these nodes: given the last graph $G_{\tau-1}$ and current graph G_τ , for all the nodes on $G_{\tau-1}$, we select flow data of the last week at $\tau - 1$ and τ . Denote these data as $X_{\tau-1}$ and X_τ , and previous SurModel $\Psi_{\tau-1}$ without the last fully connected layer is $\psi_{\tau-1}$. Patterns captured from data can be represented as the output feature U of $\psi_{\tau-1}$:

$$U_\tau = \psi_{\tau-1}(X_\tau), \quad U_{\tau-1} = \psi_{\tau-1}(X_{\tau-1}) \quad (4)$$

To tackle the problem that features are different between weekdays and weekends, in the time dimension, we normalize U_τ and $U_{\tau-1}$ and utilize histogram of the normalize features as its distribution. Let P_τ and $P_{\tau-1}$ represent these two distribution of feature value. We aim to discover the similarity between these two distribution as it reveals the change of current and previous patterns. Thus, Jensen–Shannon divergence (JSD or JS divergence) is introduced as the measurement of similarity:

$$JSD(P_\tau||P_{\tau-1}) = \frac{1}{2}D(P_\tau||\bar{P}) + \frac{1}{2}D(P_{\tau-1}||\bar{P}) \quad (5)$$

where $D(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log(P(x)/Q(x))$ is the Kullback–Leibler divergence and $\bar{P} = (P_\tau + P_{\tau-1})/2$.

If JSD of a node is pretty high, its pattern can be considered as evolving substantially. Finally, we choose the top 5% of nodes with the highest JSD score as candidate nodes for learning the new patterns. Similar to those newly added nodes, we also construct a sub-graph for GNN layers.

4.3 Historical Traffic Knowledge Consolidation

When models capture the new patterns through components mentioned above, another problem arises: the model may forget previous knowledge it has learnt because it focus on those new patterns only. The catastrophic forgetting phenomenon arouses us to consolidate historical traffic knowledge. One way to preserve existing patterns is to track back to the source where these knowledge from, i.e., historical flow data. The other is to force current training model Ψ_τ to remember useful knowledge from $\Psi_{\tau-1}$. Motivated by these two ideas, we design two components to consolidate knowledge from data and model perspectives. Details are presented as follows.

Information Replay

In order to maintain existing knowledge, we can train the model Ψ_τ with sampled historical information and make a replay. A simple strategy is to randomly sample flow data of some nodes, whereas it has large errors for these samples may be less representative. Rethinking the node detection algorithm based on JS divergence, nodes whose JSD scores are lower reveal that their flow is more stable, so that it is reasonable to choose these nodes. In the end, those nodes with the lowest JSD scores are selected and construct a sub-graph to train Ψ_τ as a replay.

Parameter Smoothing

For better preserving previous knowledge, we utilize the last model $\Psi_{\tau-1}$ to impose smoothing constraints on parameters of the current one Ψ_τ , so as to avoid large deviation of the model and the occurrence of the forgetting problem. Adopting 12 constraints is a crude approach, where excessive restrictions on the model may make it lack the ability to learn new patterns. If we know which parameters are essential to the existing patterns of traffic flows and only restrict these parameters, we can get a more expressive model for forecasting.

Therefore, we introduce elastic weight consolidation (EWC) [Kirkpatrick *et al.*, 2017], adding a more flexible and exquisite smoothing term to the loss function:

$$\mathcal{L}_{smooth} = \lambda \sum_i \mathbf{F}_i (\Psi_\tau(i) - \Psi_{\tau-1}(i))^2, \quad (6)$$

where λ is the weight of the smoothing term and \mathbf{F}_i is the importance of the i -th parameter in the model $\Psi_{\tau-1}$, which is estimated using Fisher Information as

$$\mathbf{F} = \frac{1}{|X_{\tau-1}|} \sum_{x \in X_{\tau-1}} [g(\Psi_{\tau-1}; x)g(\Psi_{\tau-1}; x)^T], \quad (7)$$

where g is the is first order derivatives of the loss.

Through the weighted smoothing term, the weight of the less essential parameters for historical data is smaller, and these parameters can better adapt to the new patterns. On the contrary, the more important parameters for historical data are more weighted, and thus their changes are restricted, ensuring the preservation of historical knowledge.

5 Experiments

5.1 Dataset

To verify the effectiveness and efficiency of our framework, we conduct some experiments on a real-world dataset, PEMS3-Stream. PEMS3-Stream is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS) [Chen *et al.*, 2001] in real-time by every 30 seconds. The data is aggregated into every 5-minutes interval from 30-seconds data instances. PEMS3-Stream contains traffic flow data in North Central Area from 2011 to 2017. We select data from July 10th to August 9th every year. The details are shown in Table 1.

5.2 Preprocessing

To construct a high-quality dataset, we select sensors by the following rules: (1) the location shifting of a sensor in corresponding metadata should be less than 100 meters; (2) the missing ratio of data of a sensor should be less than 15%; (3) the traffic network of PEMS3-Stream keeps expanding, meaning sensor i at the τ -th year must appear at the $(\tau + \Delta\tau)$ -th year, $\Delta\tau \in \{1, 2, \dots, 7 - \tau\}$.

The adjacency matrix A_τ at the τ -th year can be extracted from metadata of sensors:

$$A_\tau[mn] = \begin{cases} \exp\left(-\frac{d_{mn}^2}{\sigma_d^2}\right) & , m \neq n \text{ and } d_{mn} \leq \epsilon \\ 0 & , \text{otherwise} \end{cases} \quad (8)$$

where d_{ij} denotes the distance between sensor i and j . σ_d is set to 10 and ϵ is set to 1.0 to control the sparsity of A_τ .

5.3 Experimental Settings

Data of each year is split into training, validation and testing sets with ratio 6:2:2 in the time dimension. Our task is to forecast traffic flows in the next hour with data from the last hour as $T = K = 12$. AdamW [Loshchilov and Hutter, 2017] is

Year	2011	2012	2013	2014	2015	2016	2017
# Nodes	655	715	786	822	834	850	871
# Edges	1577	1929	2316	2536	2594	2691	2788

Table 1: The statistic of PEMSD3-Stream dataset.

PEMSD3		15 min			30 min			60 min			Running Time	
Model		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	Total	Avg. (s)
Retrained	SVR	14.76	25.06	20.16	16.15	27.36	21.73	19.02	32.24	25.13	593.42	-
	GRU	13.38	23.04	18.70	14.61	24.94	20.55	17.21	28.92	24.92	256.22	7.96
	STGCN	12.30	20.37	16.96	14.29	23.69	20.69	17.45	28.53	25.77	6518.70	147.78
	STSGCN	13.02	20.99	17.34	13.62	22.06	17.97	14.76	24.03	19.23	12076.21	162.47
	Retrained-STModel	12.76	20.59	17.83	13.92	22.76	19.90	16.33	26.90	24.99	1232.58	32.45
Online	Static-STModel	13.98	21.88	29.36	15.12	23.98	31.67	17.46	28.01	36.44	319.39	3.85
	Expansible-STModel	13.51	21.83	18.19	15.05	24.72	19.88	18.24	30.49	23.82	418.46	9.13
	TrafficStream	12.79	20.71	17.05	13.97	22.90	18.74	16.37	27.04	22.94	461.24	14.49

Table 2: Overall results of traffic forecasting on PEMS3-Stream dataset.

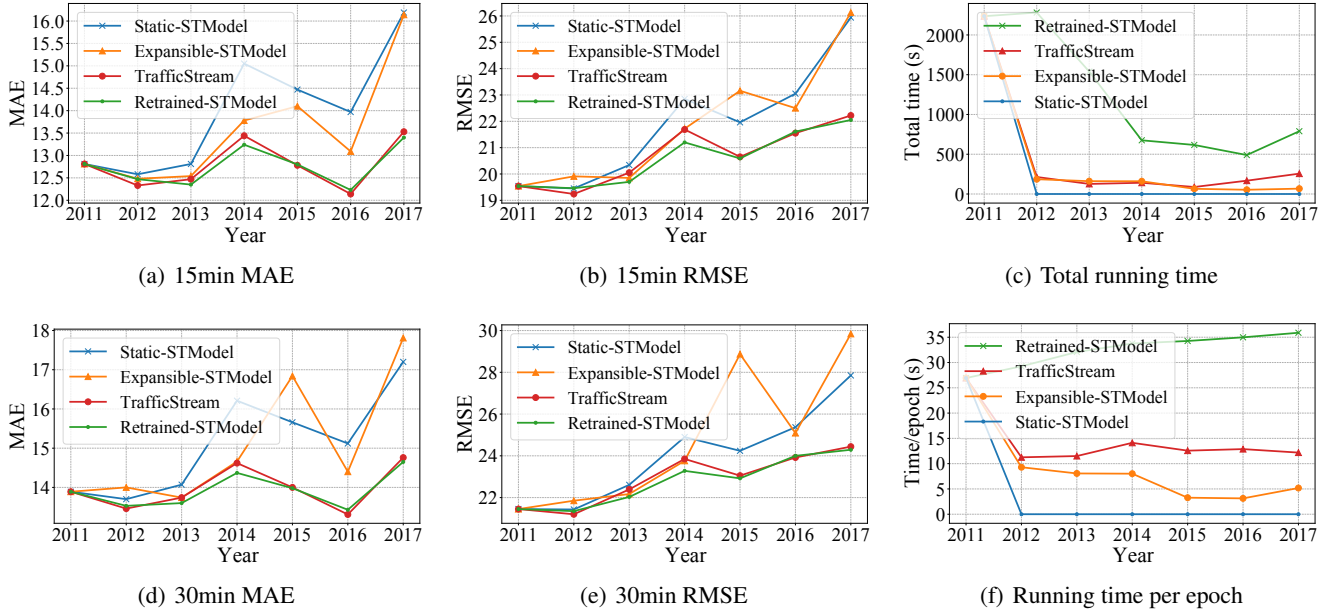


Figure 1: MAE, RMSE and running time of traffic forecasting on consecutive years.

used as the optimizer with learning rate 0.005. Batch size is 128 and we train 100 epochs for each year with early stopping to accelerate training process. We apply random search strategy to find the best hyper-parameters for TrafficStream.

To evaluate the performance of our framework, we adopt Mean Absolute Errors (MAE), Root Mean Squared Errors (RMSE) and Mean Absolute Percentage Errors (MAPE) as the metric for effectiveness. Total training time and training time per epoch are used to reveal model efficiency.

5.4 Baseline Methods

- *SVR* [Castro-Neto *et al.*, 2009]: Support Vector Regression adapts support vector machine for regression task.
- *GRU* [Cho *et al.*, 2014]: Gated Recurrent Unit is an RNN model that leverages gated mechanism.
- *STGCN* [Yu *et al.*, 2017]: Spatial-Temporal Graph Convolutional Network adopts graph convolution and gated CNN to extract spatial and temporal patterns. We train a STGCN using data from all nodes every year.

- *STSGCN* [Song *et al.*, 2020]: Spatial-Temporal Synchronous Graph Convolutional Network synchronously captures the localized spatial-temporal correlations. We train a STSGCN using data from all nodes every year.
- *Retrained-STModel*: We train a surrogate model introduced in Sec 4.1 using data from all nodes every year. We use models at the last year as initialization.
- *Static-STModel*: We utilize data at the first year (i.e., 2011) to train a surrogate model, and forecast traffic flow after 2011 using the surrogate model directly.
- *Expansible-STModel*: We train a surrogate model in an online manner, that is, at every year, we only train the data from the newly added nodes with the models trained at the last year as initialization.
- *TrafficStream*: Our proposed framework. We train a surrogate model in an online manner with new pattern fusion and historical knowledge consolidation.

5.5 Experimental Results

In this section, we compare TrafficStream with other baselines on the traffic flow forecasting task and discuss the results from effectiveness and efficiency.

Effectiveness

Table 2 shows the overall performance of traffic forecasting at different time granularity (i.e., 15-, 30-, 60-minutes in the future). We compute the averaged MAE, RMSE and MAPE in 7 years of each model. Our model, TrafficStream achieves lower prediction error compared with the traditional online training methods (i.e., Static-STModel and Expansible-STModel). And it reaches almost the same effect as the upper bound (i.e., Retrained-STModel, STGCN and STSGCN). Besides, STModel adopts GCN and CNN to capture spatial-temporal correlation. It is significantly better than simple methods like SVR and LSTM, and has comparable performance with the advanced model like STGCN.

Figure 1(a), 1(b), 1(d) and 1(e) show the effect of models on consecutive years. Every year, as Static-STModel does not update the model incrementally, it cannot capture new patterns on the traffic network, resulting in a rapid increase in prediction errors. Expansible-STModel only trains incrementally on new nodes every year and does not maintain the patterns of other nodes, leading to the problem of catastrophic forgetting. The expressive ability of the model will become unsatisfactory and unstable. Especially when the patterns of new nodes are inconsistent with other nodes (e.g., 2015), Expansible-STModel will get very poor prediction results. TrafficStream can maintain good and stable forecasting, indicating that it can continuously learn new traffic patterns while maintaining existing knowledge.

Efficiency

Table 2, Figure 1(c) and Figure 1(f) show the running time of each model, including total training time and training time per epoch, and we compute the averaged time in 7 years. The training time of Retrained-STModel and STGCN is much higher than online methods (i.e., Expansible-STModel and TrafficStream), which is about 3 times longer. It proves that TrafficStream can maintain high efficiency while achieving accurate prediction, providing the opportunity for real-world applications. Note that Static-STModel uses the model trained in the first year to predict the annual traffic flows after that, so its training time is 0 after 2011.

5.6 Ablation Study

Impact of New Pattern Fusion

We first analyze the impact of the fusion of new patterns on the forecasting performance, as shown in Figure 2. *Non-Expand* refers to not integrating the patterns of new nodes and only integrating the new patterns of existing nodes. On the contrary, *Non-Detect* represents only integrating the patterns of new nodes. *TrafficStream* integrates all possible new patterns. It shows that it is not enough only to learn any parts of new patterns. Our model fuse all potential new patterns well so as to obtain the latest and best parameters to forecast the traffic flows in the future.

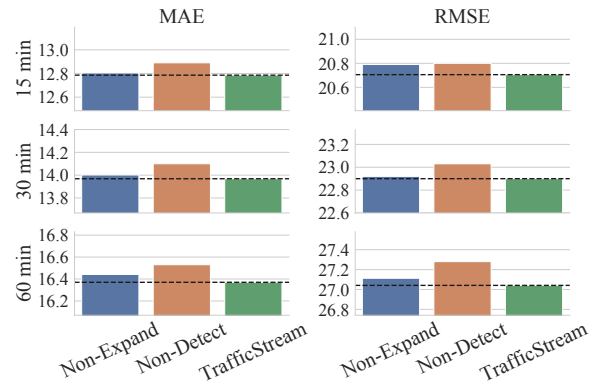


Figure 2: Ablation study of new pattern fusion.

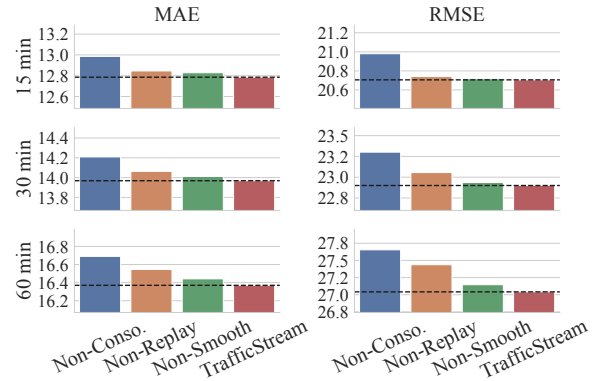


Figure 3: Ablation study of historical knowledge consolidation.

Impact of Knowledge Consolidation

We then analyze the influence of historical knowledge on the forecasting performance, as shown in Figure 3. *Non-Conso.* represents that it does not consolidate any knowledge learned by the previous model at all. *Non-Replay* and *Non-Smooth* respectively refer to training without data replay and without model smoothing. *TrafficStream* utilizes both approaches simultaneously. The results of not consolidating historical knowledge are the worst, indicating that the forgetting of historical knowledge occurs, so that the prediction of some nodes becomes worse. The two consolidation approaches can avoid the forgetting problem to a certain extent, and the approach of data replay is better. Besides, the combination of the two further improves the predictive ability of the model.

6 Conclusion

In this paper, we propose a GNN framework for streaming traffic flow forecasting. Continual learning is introduced to achieve efficient update and effective prediction in streaming traffic data. The model is evaluated and proved to be very effective on PEMS3-Dynamic dataset. Compared with the traditional retraining approach, TrafficStream achieves high prediction accuracy while reducing training complexity greatly.

References

- [Castro-Neto *et al.*, 2009] Manoel Castro-Neto, Young-Seon Jeong, Myong-Kee Jeong, and Lee D Han. Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert systems with applications*, 36(3):6164–6173, 2009.
- [Chen *et al.*, 2001] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001.
- [Chen *et al.*, 2020] Xu Chen, Yuanxing Zhang, Lun Du, Zheng Fang, Yi Ren, Kaigui Bian, and Kunqing Xie. Tssrgcn: Temporal spectral spatial retrieval graph convolutional network for traffic flow forecasting. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 954–959, 2020.
- [Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [Guo *et al.*, 2019] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.
- [Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- [Li *et al.*, 2017] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.
- [Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [Mengzhang and Zhanxing, 2020] Li Mengzhang and Zhu Zhanxing. Spatial-temporal fusion graph neural networks for traffic flow forecasting. *arXiv preprint arXiv:2012.09641*, 2020.
- [Rolnick *et al.*, 2019] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 348–358, 2019.
- [Rusu *et al.*, 2016] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- [Song *et al.*, 2020] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 914–921, 2020.
- [Wang *et al.*, 2020a] Chen Wang, Yuheng Qiu, and Sebastian Scherer. Lifelong graph learning. *arXiv preprint arXiv:2009.00647*, 2020.
- [Wang *et al.*, 2020b] Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. Streaming graph neural networks via continual learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1515–1524, 2020.
- [Wang *et al.*, 2020c] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, pages 1082–1092, 2020.
- [Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.
- [Yao *et al.*, 2018] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [Yu *et al.*, 2017] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.
- [Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [Zhou *et al.*, 2020] Fan Zhou, Chengtai Cao, Ting Zhong, Kunpeng Zhang, Goce Trajcevski, and Ji Geng. Continual graph learning. *arXiv preprint arXiv:2003.09908*, 2020.
- [Zivot and Wang, 2006] Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®*, pages 385–429, 2006.