# ADA-STNET: A DYNAMIC ADABOOST SPATIO-TEMPORAL NETWORK FOR TRAFFIC FLOW PREDICTION

*Jiawei Sun*[1], *Jie Li*[1,2,*], *Chentao Wu*[1], *Zili Tang*[1], *Celimuge Wu*[3]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
[3]Graduate School of Informatics and Engineering, The University of Electro-Communications, Japan

## ABSTRACT

Traffic flow prediction is of particular interest since its massive applications in intelligent transportation systems (ITS). The problem is challenging due to the complex spatio-temporal correlations and nonlinearities of traffic flows. However, existing methods based on the graph neural networks cannot efficiently extract the dynamic and long-range spatial correlations, thus producing unsatisfactory prediction results. In this paper, we propose an AdaBoost Spatio-temporal Network (Ada-STNet). Similar to AdaBoost, Ada-STNet stacks several base neural networks as "layers" which capture spatial and temporal correlations simultaneously. Each layer learns an adaptive adjacency matrix from weights and embedding of nodes. The adjacency matrix is layer-wise adjusted to extract information from distant neighbors and adapt to dynamic correlations. Experiments are conducted on three real-world benchmark datasets, demonstrating that the Ada-STNet outperforms the state-of-the-art methods.

***Index Terms***— Traffic prediction, road sensor network, graph neural network, signal processing over graphs

## 1. INTRODUCTION

With the rapid development of urbanization and the expansion of traffic demands, transportation planning is growingly essential for people's daily life and government management [1]. As a crucial component of the intelligent transportation system (ITS), traffic prediction can help city decision-makers accomplish transportation planning tasks better. A tremendous amount of traffic data, including vehicle flows, taxi demands, and GPS trajectories, are collected and analyzed for traffic prediction tasks. The traffic management department can direct vehicles reasonably in advance to improve traffic efficiency and relieve congestion by predicting future traffic.

The traffic prediction problem can be naturally formulated as multivariate time series forecasting over a graph. One of the biggest challenges is the high correlations among variables due to the connectivity of roads or closeness between sensors [2]. Statistical methods for the univariate time series prediction such as vector auto-regression (VAR) [3] and auto-regressive integrated moving average (ARIMA) [4] cannot scale well to multivariate time series due to the stationary and independence assumptions. The deep learning models have advantages of automatic feature representation learning and excellent performance with big data [5]. Temporal correlations are captured with Recurrent Neural Network in [6–8] or attention mechanism in [9, 10]. Graph neural networks (GNN) are ideal for the traffic prediction problem since GNN can capture spatial dependencies by aggregating neighbor nodes' information [11]. With graph signals as the input, tremendous GNN-based models [12–15] have shown superior performance to traditional deep-learning methods on various traffic data problems. However, existing spatio-temporal models are mainly based on shallow GNN, which is limited to the over-smoothing problem [16]. These models cannot aggregate information from high-order neighbor nodes, i.e., distant sensors in road networks. Besides, these models ignore the dynamic property of spatial correlations caused by varying real-time traffic conditions. Moreover, the mean absolute error over all nodes is used as the loss function, resulting a large variation in prediction errors across nodes.

To overcome the above limitations, we propose a novel ***Ada****Boost* ***S****patio-****T****emporal* ***Net****work* (Ada-STNet). Ada-STNet stacks several base predictors as layers that incorporate graph convolution networks in the gated recurrent unit to capture the hidden spatio-temporal patterns efficiently. Each layer generates a self-adaptive adjacency matrix based on the weights and embedding of nodes. After each layer is trained in sequence, the weights of nodes are updated dynamically based on the node-wise prediction results with the validation set. Our main contributions are as follows,

- We are the first to introduce the ensemble learning concept into spatio-temporal prediction models and develop a novel RNN-like model making the prediction results more balanced and accurate.
- Extensive experiments are conducted on three benchmark traffic datasets to evaluate the proposed method. The results show that our model outperforms all the

*Corresponding author

baseline methods on all traffic datasets.

## 2. PROBLEM FORMULATION

First, define an undirected sensor network $G = (V, E, A)$, where $|V| = N$ is the set of sensor nodes, $E$ is the set of edges representing connectivity of nodes, and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix containing edge information of graph. The sensor network can be observed as a graph signal, denoted as $X_G = \{X_{1,:}, X_{2,:}, ..., X_{N,:}\} \in \mathbb{R}^{N \times T \times C}$, where $X_{n,:} = \{x_{n,1}, x_{n,2}, \cdots x_{n,T}\} \in \mathbb{R}^{T \times C}$ is the traffic flow record observed by node $n$, $T$ is the length of time series, and $C$ is the number of features (e.g., speed, flow). The graph-based traffic prediction problem can be formulated as learning a function $F(\cdot)$ which at one time step $t$ predicts values in the next $H$ time steps given the previous $H'$ historical values:

$$\left[X_G^{(t-H'+1)}, \cdots, X_G^{(t)}\right] \xrightarrow{F(\cdot)} \left[X_G^{(t+1)}, \cdots, X_G^{(t+H)}\right].$$

## 3. METHODOLOGY

### 3.1. General Architecture

In this section, we elaborate on the framework of the proposed Ada-STNet model. We aim to integrate AdaBoost into the spatio-temporal graph network to capture dynamic spatial correlations and to moderate imbalanced prediction errors among nodes. As illustrated in Fig.1, unlike most graph neural networks with various structures for each layer, Ada-STNet consists of multiple base predictors with the same architecture as "layers." Each "layer" first generates a self-adaptive adjacency matrix based on the weights and embedding of nodes. Then each layer is trained sequentially, followed by increasing weights of nodes with poorly predicted results and decreasing weights of nodes with well-predicted ones. Finally, the outputs from different base predictors are combined by selecting the best predictor for each node or fully connected layers. Two key innovations of Ada-STNet compared to traditional AdaBoost are: 1) Ada-STNet only defines one base predictor which is repeatedly optimized. 2) Ada-STNet adjusts weights of nodes during training instead of weights of samples in traditional AdaBoost.

### 3.2. Base Predictor

We establish the base predictor graph convolution recurrent network (GCRN) as follows.

**Graph Generation Layer.** In most GNN-based spatio-temporal models, a pre-defined adjacency matrix is needed and defined depending on distance [13] or similarity (e.g., POI information [17], DTW similarity [18]) of nodes. In Ada-STNet, we use the weighted adaptive adjacency matrix, which is learned from data during the training process. First, define a learnable node embedding with random initialization $\hat{E} \in \mathbb{R}^{N \times D}$ with embedding dimension $D$. Then

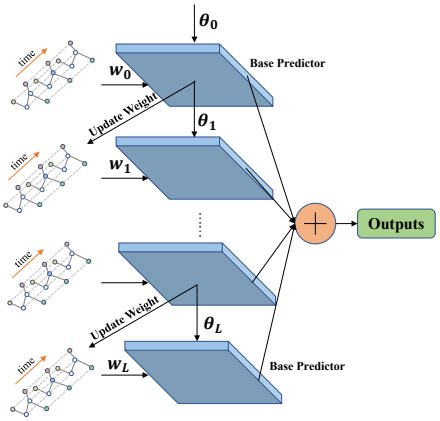


**Fig. 1**. The overview of Ada-STNet. Each base predictor has the same network design. The parameter of one base predictor will be conveyed to the next one after training, while the weights of nodes will be updated.

the weighted adaptive adjacency matrix showing the hidden spatial correlation is:

$$\hat{A} = softmax(LeakyRelu(\hat{E}\hat{E}^T \odot WW^T)), \quad (1)$$

where $W \in \mathbb{R}^N$ are the weights of nodes, the element-wise product of $\hat{E}\hat{E}^T$ and $WW^T$ measures the weighted similarity among nodes, the $LeakyRelu()$ is the activation function to add nonlinearity, and the $softmax()$ normalizes the element of adjacency matrix to $(0, 1)$ range.

**Graph Convolution Recurrent Network.** Gated Recurrent Unit (GRU) [19] can only handle the temporal correlations since the network's connectivity is neglected. Thus, in our base predictor GCRN, the multi-layer perceptron in GRU is replaced by graph convolution network (GCN) [11] to extract correlation information in the spatial domain, as illustrated in Fig.2. The graph convolution layer, which aggregates information from each node's neighbors, is defined as:

$$Z = AX\Theta, \quad (2)$$

where $A \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix, $X \in \mathbb{R}^{N \times C}$ delegates the input graph signal, and $\Theta \in \mathbb{R}^{C \times D}$ stands for the learnable parameters.

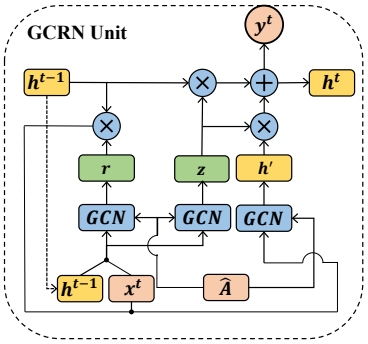


**Fig. 2**. The architecture of base predictor GCRN.

Given the learned adjacency matrix $\hat{A}$, the current input $x^t$, and the hidden state $h^{t-1}$ passed from the previous time

step, which contains historical information, GCRN will get the current output $h^t$ which is passed to the next time step. By substituting Eq.2, the formula of GCRN is as follows:

$$r = \sigma(\hat{A}[X_G^t, h_{t-1}]W_r + b_r) \tag{3}$$

$$z = \sigma(\hat{A}[X_G^t, h_{t-1}]W_z + b_z) \tag{4}$$

$$h' = \tanh(\hat{A}[X_G^t, r \odot h^{t-1}]W_{h'} + b_{h'}) \tag{5}$$

$$h^t = (1 - z) \odot h^{t-1} + z \odot h', \tag{6}$$

where $X_G^t$ is the input graph signal at time $t$, $[\cdot]$ is the concatenation operation, $\odot$ is the Hadamard product, $r$ and $z$ are states of the reset and update gate respectively, $h'$ is the state of the candidate activation, $W_r, W_z, W_{h'}, b_z, b_r$ , and $b_{h'}$ are the learnable parameters.

### 3.3. Training and Inference

**Loss Function.** Taking nodes' weights into consideration, we utilize the weighted mean absolute error (WMAE) as the optimization objective for improving the poorly performed nodes:

$$L(\hat{Y}, Y) = \frac{1}{TN} \sum_{i=1}^{T} \sum_{j=1}^{N} w_j \cdot |\hat{Y}_{j,t+i} - Y_{j,t+i}|, \tag{7}$$

where $w_i$ is the weight of each node.

**Training and Inference of Ada-STNet.** We propose the training algorithm for the Ada-STNet model referring to AdaBoost.RT [20] as described in Algorithm 1. A list to record the best base predictor for each node is maintained during the training process i.e., which base predictor achieves the best prediction results on the validation set for each node. In the inference period first input the test data into $L$ base predictors to obtain $L$ groups of outputs $Y_G^l \in \mathbb{R}^{N \times H}$. The final output is the combination of the predictions on each node from the node's best predictor . For example, the second base predictor is the best predictor for the first node, then select the first column of $Y_G^2$ as the first column of the final prediction result.

## 4. EXPERIMENT

### 4.1. Datasets

We evaluate our model on three benchmark traffic datasets: PeMS03, PeMS04, PeMS07 [21] collected from Caltrans Performance Measure System (PeMS), which measures the highway traffic every 30 seconds in California. The detailed information of datasets is shown in Table 1. Three datasets are aggregated into 5-minutes windows. We adopt identical data pre-processing techniques as in [22]. One-hour historical data is used to predict the next hour's data, i.e., using the past 12 continuous time steps to predict the future 12 time steps. The datasets are split in chronological order with 60% for training, 20% for validation, and 20% for testing.

**Algorithm 1** Ada-STNet Training Algorithm

**Input**: The graph signal dataset $X$, the graph $G = (V, E)$, the base predictor GCRN $f_\theta(\cdot)$, a list $s$ to record the best predictor for each node, the number of base predictors $L$.

1: Initialize the weights of nodes $w_i^0 = 1/|V|$ for each node. Randomly initialize all $L$ base predictors $f_\theta^l(\cdot)$.
2: **for** $l$ = 0 to L **do**
3: Train the base predictor $f_\theta^l(\cdot)$ with graph signal $X_G$ by minimizing the weighted mean absolute error in Eq 7.
4: Compute the mean absolute error $e_i^l$ for each node $i$ with base predictor $f_\theta^l(\cdot)$ over the validation set
5: Update the list $s$ if $e_i^l$ is smaller than the error of previous best base predictor on some nodes.
6: Normalize $e_i^l$ with a standard core and update the weight for each node $i$ as:

$$w_i \leftarrow \frac{w_i}{Z_l} \cdot exp(\frac{e_i^l - mean(e^l)}{std(e^l)})$$

where $Z_t$ is a normalization factor chosen such that $w_i$ will be a distribution.
7: If $l < L$, copy the parameter $\theta$ of $f_\theta^l(\cdot)$ to $f_\theta^{l+1}(\cdot)$
8: **end for**

**Table 1**. Summary of Dataset

| Name | Region | # Nodes | #Edges | #Samples |
|---|---|---|---|---|
| PeMS03 | North Central | 358 | 547 | 26208 |
| PeMS04 | Bay Area | 307 | 340 | 16992 |
| PeMS07 | Los Angeles | 883 | 866 | 28224 |

### 4.2. Experimental Settings

**Baselines.** We compare Ada-STNet with statistical methods, traditional deep learning models, and recently state-of-the-art graph-based models, including **VAR** [3]: Vector Auto-Regression. **ARIMA** [4]: Auto-Regressive Integrated Moving Average method. **LSTM** [23]: Long Short-Term Memory network. **STGCN** [13]: Spatio-temporal graph neural network first utilized the GCN to capture the spatial correlation. **ASTGCN** [14]: Attention-based spatio-temporal graph convolution network incorporates attention mechanisms on top of STGCN. **STSGCN** [24]: Spatial-temporal Synchronous Graph Convolutions Networks stacks multiple two-layer GCN to generate predictions of each time step. **AGCRN** [25]: Adaptive Graph Convolutional Recurrent Network generates a graph with learnable embedding.

**Parameter setup.** We implement the Ada-STNet model using PyTorch 1.9.0 [26]. The same hyper-parameter setting is applied on three datasets. The number of base predictors is set to 10. The embedding dimension in the graph generation layer is 12. The batch size is set to 64. And the hidden unit for each base predictor is set to 64. All models are optimized with Adam optimizer with learning rate 0.003. Additionally, early stop strategy is applied on all models with the patience of 15. We choose three metrics to evaluate models including mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE).

**Table 2**. Performance comparison among different approaches for traffic flow prediction.

| Datasets | Metrics | ARIMA | VAR | LSTM | STGCN | ASTGCN | STSGCN | AGCRN | **Ada-STNet** |
|---|---|---|---|---|---|---|---|---|---|
| PeMS03 | RMSE | 47.59 | 38.26 | 35.51 | 30.42 | 29.56 | 29.21 | 28.91 | **27.32** |
| | MAE | 35.41 | 23.69 | 21.45 | 17.61 | 17.87 | 17.48 | 16.48 | **15.77** |
| | MAPE | 33.78% | 24.51% | 25.12% | 17.62% | 20.40% | 16.78% | 16.19% | **15.20%** |
| PeMS04 | RMSE | 48.81 | 43.78 | 41.69 | 38.95 | 35.42 | 34.18 | 33.12 | **31.59** |
| | MAE | 33.73 | 29.75 | 27.23 | 26.13 | 23.65 | 22.53 | 20.62 | **19.45** |
| | MAPE | 24.18% | 20.09% | 18.51% | 16.67% | 16.72% | 14.12% | 13.69% | **13.04%** |
| PeMS07 | RMSE | 59.27 | 115.24 | 45.72 | 39.42 | 44.62 | 40.31 | 38.96 | **37.36** |
| | MAE | 38.17 | 75.36 | 30.23 | 25.29 | 29.75 | 24.83 | 23.98 | **23.37** |
| | MAPE | 19.46% | 32.22% | 13.41% | 11.12% | 14.58% | 11.06% | 10.34% | **9.96%** |

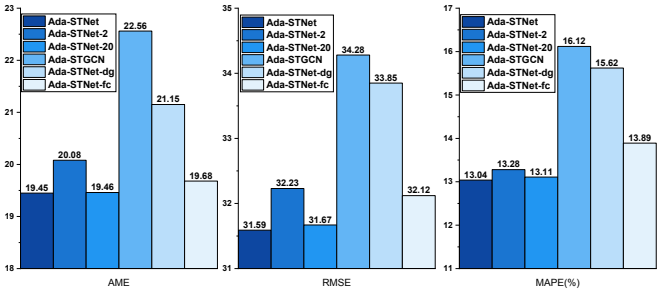


**Fig. 3**. Ablation study Results on PeMS04 dataset.

### 4.3. Performance Comparison and Analysis

Table 2 compares the predictions performance of Ada-STNet and baseline models in average RMSE, AME, MAPE over 12 horizons on three PeMS datasets. Our Ada-STNet outperforms baselines in three datasets on all metrics. Specifically, statistic methods ARIMA, VAR, and traditional deep learning model LSTM do not perform well since they ignore the spatial correlations. By contrast, GNN-based models achieve better performance by taking advantage of spatial information of the road network, which demonstrates the importance of hidden spatial patterns in the traffic forecasting problem. Compared with other GNN-based models, the superiority of Ada-STNet can be attributed to the following three aspects,

1. Ada-STNet dynamically adjusts the weights of nodes enabling each base predictor to focus more on optimizing the prediction of underperforming nodes.
2. A new adaptive graph is learned for each base predictor to adapt to the dynamic characteristic of correlations.
3. The node-wise errors of a single predictor will be compensated by other base predictors. Thus, the combined prediction performance of the whole Ada-STNet would be superior to a single predictor.

### 4.4. Ablation Study

To further investigate the proposed Ada-STNet model, we compare the Ada-STNet with its five variants, where Ada-STNet-2 and Ada-STNet-20 contain two and twenty base predictors respectively, Ada-STGCN replaces GCRN with STGCN [13], Ada-STNet-dg removes the graph learning layer and utilizes the distance-based graph, Ada-STNet-fc combined base predictors with fully connected layers.

Fig.3 reports the comparison result of different variants, from which we have the following observations. First, the fact that Ada-STNet with two layers outperforms single base predictors verifies that stacking layers in our approach can improve performance significantly. Stacking more base predictors can obtain better results when the number of base predictors is less than 9. Second, Ada-STNet outperforms Ada-STGCN showing that the selection of base predictors affects the performance of our model. Supplementarily, compared to Ada-STNet which improves the base model by 5%, Ada-STGCN improves the base model by 12%, which indicates that stacking base predictors like our method improves more on simpler and worse-performed base models. Third, Ada-STNet-dg has a poor performance, which shows the efficiency of the dynamically-adjusted adaptive graph learned from the embedding of traffic data. Lastly, Ada-GCRN-fc, which utilizes a fully connected layer to combine base predictors, has a similar performance with Ada-STNet but requires more training time and GPU memory. Thus we keep a simple way which selects the appropriate base predictor for each node to combine base models.

## 5. CONCLUSION

In this paper, we propose a novel spatio-temporal forecasting model named Ada-STNet. The Ada-STNet stacks multiple base predictors as "layers" in an AdaBoost-like way. All base predictors share the same architecture, which replaces the multi-layer perceptron of GRU with the graph convolution network to capture spatio-temporal correlation. Meanwhile, the adaptive adjacency matrix and weights of nodes are dynamically adjusted in each base predictor. The weighted average mean error is proposed as the loss function to concentrate on nodes with inaccurate prediction results. Experiments on three real-world datasets show that Ada-STNet outperforms state-of-the-art models for traffic data prediction problems.

## 6. ACKNOWLEDGEMENT

This work has been partially supported by the National Key Research and Development Program of China Nos. 2020YFB1806700, 2020YFB1710900, NSFC Grant 61932014, Project BE2020026 supported by the Key R&D Program of Jiangsu, China.

## 7. REFERENCES


[1] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[2] Weiwei Jiang and Jiayun Luo, "Graph neural network for traffic forecasting: A survey," *arXiv preprint arXiv:2101.11174*, 2021.

[3] Eric Zivot and Jiahui Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-Plus®*, pp. 385–429, 2006.

[4] Billy M Williams and Lester A Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[5] Senzhang Wang, Jiannong Cao, and Philip Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE transactions on knowledge and data engineering*, 2020.

[6] Ata Akbari Asanjan, Tiantian Yang, Kuolin Hsu, Soroosh Sorooshian, Junqiang Lin, and Qidong Peng, "Short-term precipitation forecast based on the persiann system and lstm recurrent neural networks," *Journal of Geophysical Research: Atmospheres*, vol. 123, no. 22, pp. 12–543, 2018.

[7] Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes, "Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection," *Neural networks*, vol. 108, pp. 466–478, 2018.

[8] Yuecheng Rong, Zhimian Xu, Ruibo Yan, and Xu Ma, "Du-parking: Spatio-temporal big data tells you realtime parking availability," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 646–654.

[9] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1459–1468.

[10] Xian Zhou, Yanyan Shen, Yanmin Zhu, and Linpeng Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 736–744.

[11] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[12] 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.

[13] 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.

[14] 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*, 2019, vol. 33, pp. 922–929.

[15] Xiyue Zhang, Chao Huang, Yong Xu, and Lianghao Xia, "Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1853–1862.

[16] Meng Liu, Hongyang Gao, and Shuiwang Ji, "Towards deeper graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 338–348.

[17] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, pp. 3656–3663.

[18] Mengzhang Li and Zhanxing Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *arXiv preprint arXiv:2012.09641*, 2020.

[19] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[20] Dimitri P Solomatine and Durga L Shrestha, "Adaboost. rt: a boosting algorithm for regression problems," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*. IEEE, 2004, vol. 2, pp. 1163–1168.

[21] 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*, 2019, vol. 33, pp. 922–929.

[22] 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*, 2020, vol. 34, pp. 914–921.

[23] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] 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*, 2020, vol. 34, pp. 914–921.

[25] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *arXiv preprint arXiv:2007.02842*, 2020.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.