FlowDiviner: A Spatio-Temporal Network Traffic Prediction Method Based on Graph Neural Network

Liang Qin
Xidian University
liang.qin.xidian@foxmail.com

Wenting Wei
Xidian University
wtwei@xidian.edu.cn

Yinhao Ma
Xidian University
xd.mayinghao@gmail.com

ACM Reference Format:

Liang Qin, Wenting Wei, and Yinhao Ma. 2021. FlowDiviner: A Spatio-Temporal Network Traffic Prediction Method Based on Graph Neural Network. In *5th Asia-Pacific Workshop on Networking (APNet 2021) (APNet 2021), June 24–25, 2021, Shenzhen, China, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3469393.3469671

1 INTRODUCTION

End-to-end traffic prediction plays an important role in network management. However, network traffic volume is hard to forecast due to its characteristics of non-linear, abrupt, and time-sensitive.

The existing works mainly contain linear and non-linear traffic prediction methods.The liner traffic predictors represented by ARIMA are mainly effective for short term traffic but not work well for long term traffic. Thanks to its powerful non-linear representation ability, deep learning[4] is used by researchers to predict the future network traffic successfully[2][1]. However, most of existing works only consider the temporal correlation of traffic and ignore the spatial correlation of network traffic, resulting in poor accuracy. The main reason is that network traffic itself has strong spatial correlation as well as temporal correlation. The burst traffic of a node in the network will have an impact on the traffic trend of the surrounding nodes in addition to the traffic trend of this node at the next moment.

We proposed a new long term network traffic prediction method FlowDiviner. FlowDiviner is an encoder and decoder architecture. The GCN model is used to capture the spatial characteristics of traffic from the network topology. The Multi-Head Attention layer is used to extract the dynamic trend of network traffic to obtain temporal characteristics. At the same time, theres a Middle Attention Block between the encoder and the decoder, determine the relationship between the historical and the future state of traffic by using the mechanism of Multi-Head Attention, therefore, reduce the influence of the cumulative error of the model by the way. Preliminary experiments on real Abilene network traffic dataset show that FlowDiviner has outstanding performance in long term traffic prediction.

APNet 2021, June 24–25, 2021, Shenzhen, China, China

ACM ISBN 978-1-4503-8587-9/21/06…$15.00
https://doi.org/10.1145/3469393.3469671

2 TRAFFIC PREDICTION MODEL

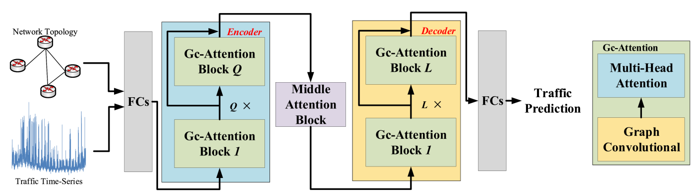

Figure 1: overview framework of FlowDiviner.

Spatial correlation extraction: Graph convolutional neural network can effectively process non-Euclidean data such as graph. Since the network topology can be modeled as a graph, we use the graph convolutional neural network to extract the spatial characteristics of traffic data. A GCN model contains multiple convolutional layers, and its feature propagation rule can be expressed as:

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

where, $\tilde{A}$=A+I_N is the adjacency matrix that adds self-connection, $A \in R^{N\times N}$ is the adjacency matrix, I_N is the identity matrix, $W^{(l)}$ is the weight matrix of layer l, $H^{(l)}$ is the output of layer l, and $H^{(0)} = X$, $X \in R^{N\times P}$ is the flow characteristic matrix, $\sigma(\bullet)$ is the activation function. Here, we choose a GCN model that contains two layers of convolution layer for processing graph data, which can be expressed as:

$$Z = f(X, A) = \sigma(\widehat{A}\mathrm{ReLU}(\widehat{A}XW^{(0)})W^{(1)}) \tag{2}$$

where,

$$Z = f(X, A) \in R^{N\times K}$$

is the output feature of K step lengths predicted. $\widehat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ is the pre-processing step, $\tilde{D}$ is the node degree matrix, $W^{(0)} \in$

$R^{P\times H}$ is the weight of input layer to hidden layer, $W^{(1)} \in R^{H\times K}$ is the weight of hidden layer to output layer, σ and $ReLU$ are activation functions.

Temporal correlation extraction: We use the multi-head attention mechanism to model the relationship between current and past time step traffic states.

$$\alpha_{t,t_j}^c = soft\max(e_{t,t_j}^c) \tag{3}$$

where, α_{t,t_j}^c is the attention score in the c^{th} head, indicating the importance of time step t to t_j. e_{t,t_j}^c represents the correlation between time step t and t_j in the c^{th} head, and the calculation method is as follows:

$$e_{t,t_j}^c = \frac{A \bullet \left\langle W_q h_{v_i,t}^l, W_k h_{v_i,t_j}^l \right\rangle}{\sqrt{d_c}} \tag{4}$$

where, $h_{v_i,t}$ and h_{v_i,t_j} are respectively the hidden states of nodes in time step t to t_j, W_q and W_k are the trainable parameter matrix, A is the adjacency matrix, and d_c is the scaling factor of the c^{th} head.

After getting the attention score, the node v_i hidden state in the time step t_j can be updated as:

$$h_{v_i,t_j}^l = \|_{c=1}^C \left\{ \sum\nolimits_{t\in N_{t_j}} \alpha_{t,t_j}^c \bullet W_v \bullet h_{v_i,t}^l \right\} \tag{5}$$

where, $\|$ represents concatenation operationW_v represents value matrix.

Middle Attention Block: The cumulative error increases with the increase of the predicted time step. We added a Middle Attention Block between the encoder and the decoder to model the past and future time steps of the traffic state. The module includes a multi-head attention layer, which uses the encoded traffic characteristics as input to the decoder to further extend the ability of the model to focus on a certain state. For any node in the network, the correlation between future time step t_k and historical time step t_p can be expressed as:

$$\beta_{t,t_j}^c = soft\max(e_{t,t_j}^c) \tag{6}$$

$$e_{t,t_j}^c = \frac{A \bullet \left\langle W_q^c h_{v_i,t}^l, W_k^c h_{v_i,t_j}^l \right\rangle}{\sqrt{d_c}} \tag{7}$$

After the Attention score is calculated, the correlation between the traffic status of the time step P after historical coding can be obtained, as shown in Equation8 :

$$h_{v_i,t_j}^l = \|_{c=1}^C \left\{ \sum\nolimits_{t=1}^P \beta_{t,t_j}^c \bullet W_v^c \bullet h_{v_i,t}^l \right\} \tag{8}$$

3 PRELIMINARY EVALUATION

We train and evaluate FlowDiviner based on a real traffic dataset collected from Abilene Network[3]. The Abilene network consists of 12 nodes connected by 13 links. At the same time, in Abilene traffic data set, each Timestep within defined as 5 minutes, to smooth the data distribution and speed up the model convergence, we use maximum minimum normalization methods for processing the raw data.

Root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) are used to evaluate the performance of FlowDiviner. The better the performance, the closer the values of these three indexes are to 0. We choose ARIMA, LSTM and GRU as Baseline methods to compare with FlowDiviner in terms of performance.

Table 1: Evaluation Result

Time	Metrics	ARIMA	LSTM	GRU	FlowDiviner
15min	RMSE	415.36	324.67	316.48	**238.47**
	MAE	315.86	164.55	158.99	**125.41**
	MAPE	7.43%	5.68%	**5.19%**	5.48%
30min	RMSE	653.95	449.19	439.48	**407.16**
	MAE	346.52	199.37	199.26	**184.34**
	MAPE	9.18%	6.89%	6.72%	**6.24%**
60min	RMSE	885.95	565.58	547.02	**486.58**
	MAE	420.67	266.75	249.87	**224.57**
	MAPE	13.92%	8.97%	8.75%	**7.48%**

Table1 shows the results of each index in 15 minutes (3 steps), 30 minutes (6 steps) and 60 minutes (12 steps) prediction on the traffic data set generated by the Abilene network with different schemes.we can find that FlowDiviner has achieved better performance in all time domains, especially in the long-term prediction, with the maximum improvement of about 15%, which proves the necessity of considering spatial correlation in the flow prediction task.

4 CONCLUSION

In this poster, we propose a new network traffic prediction method based on Graph neural network . Preliminary experiments show that FlowDiviner is effective. In the future, we will further verify the performance of FlowDiviner in larger scale networks, such as Geant23, Gemany50, etc.

REFERENCES

[1] Yingqi L, Juan W, and et al. 2020. Smoothing-Aided Support Vector Machine Based Nonstationary Video Traffic Prediction Towards B5G Networks. *IEEE Trans. Veh. Technol.* (2020).

[2] Laisen N, Xiaojie W, and et al. 2021. Network Traffic Prediction in Industrial Internet of Things Backbone Networks: A Multi-Task Learning Mechanism. *IEEE Trans Industr Inform* (2021).

[3] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. 2010. SNDlib 1.0–Survivable Network Design Library. *Networks* 3 (2010).

[4] Helin Y, Xianzhong X, and et al. 2020. Machine Learning Techniques and A Case Study for Intelligent Wireless Networks. *IEEE Netw* (2020).