Bandwidth Resources Prediction for Dynamic Networking Slicing Based on Time Series Analysis

1st Haoxian Feng
Computer and Information College
Fujian Agriculture and Forestry University
Fuzhou, China
fenghx@fafu.edu.cn

2nd Zhaogang Shu*
Computer and Information College
Fujian Agriculture and Forestry University
Fuzhou, China
zgshu@fafu.edu.cn

3rd Zhifang Zhang
Computer and Information College
Fujian Agriculture and Forestry University
Fuzhou, China
zhangzhifang@fafu.edu.cn

4th Yuantao Wang
Computer and Information College
Fujian Agriculture and Forestry University
Fuzhou, China
ytwang@fafu.edu.cn

5th Zhiwei Liu
Computer and Information College
Fujian Agriculture and Forestry University
Fuzhou, China
liuzhiwei@fafu.edu.cn

***Abstract*—To address diversified service requirements of the users, researchers proposed the concept of network slicing (NS) that can enhance the reliability and security of the services through customizing resources according to specific requirements. In the scenario of dynamic slicing, predicting the bandwidth resources requirements can help network providers adjust the resources that have been allocated in better ways. In our research, we analyze the network traffic data based on time series analysis and evaluate the performance of the state-of-art methods through two novel metrics. The experiment results show that the network operator should select the prediction method according to the slices' service type in order to balance the profits of network providers and the experience quality of slices.**

Keywords-dynamic slicing; time series analysis; bandwidth resources prediction

I. INTRODUCTION

As we are entering the Internet of Everything (IoE) era, the requirements of network use cases are very diverse. However, the current network communication architecture employs a one-size-fits-all approach to supply services, which disregards the vertical requirements of use cases. Hence, it is crucial to explore new techniques to address the diversified service requirements.

With the developments of SDN and NFV, NGMN propose the concept of network slicing to meet the different performance requirements that verticals impose in terms of latency, scalability, availability, and reliability. A network slicing is an end-to-end logical network running on the shared physical infrastructure. Network slices are customized by specific use cases and isolated from each other, so they can enhance the reliability and security of each use case and optimize the allocation of physical network resources [1]. According to the life cycle of slices, the researchers divide the slices into two types: permanent slices and temporary slices. To improve the utilization of physical resources and decrease the probability of Service Level Agreement (SLA) violations further, network providers should adjust the resources that have been allocated based on the temporal variation of the resource requirements of permanent slices, or temporary slices with long service time. This concept is defined as dynamic slicing [2]. But currently, the related works assume the future resources requirements or the distribution of resources requirements are known [3], and some works are conducted based on resources uncertainty [4].

In practice, the temporal variation of network traffic is regular. For example, the utilization of link resources is higher during the day than at night [5], which means that the bandwidth resources requirements for some slices are higher during the day than at night, and this is the well-known diurnal pattern [6]. To adjust the resources that have been allocated more purposely, it is necessary to design an effective prediction mechanism to determine the demand of future bandwidth resources for the slices. As presented in Fig. 1, the temporal variation of bandwidth resources for a permanent slice is periodic, the red line represents the fixed bandwidth resources that the network providers allocate to this slice. In the scenario of dynamic slicing, during the dawn, we hope to adjust the fixed allocated resources to the green line which is determined by the prediction mechanism. Similarly, we can also adjust the allocated resources during the day.

Because the bandwidth resources can affect how many bytes a slice can transmit at a time, and it is difficult to gather the time series data of bandwidth resources requirements while the time series data of bytes can be collected easily, we define our task as the multi-step bytes prediction problem. The prediction result can represent the future bandwidth resources requirements to some extent.

In this paper, we solve the task based on time series analysis theory, and we define two novel evaluation metrics P, Q, to measure the prediction performance of existing time series prediction methods [7-9], where P is applied to measure the extent to which the predicted value meets the bandwidth resources requirements of a slice, Q

978-1-6654-9858-6/22/$31.00 ©2022 IEEE
DOI 10.1109/MLCCIM55934.2022.00009

is applied to measure the extent to which the predicted value can save bandwidth resources.

This paper is organized as follows. Section II introduce the related work. Section III discuss the details of the dataset used and the data preprocessing. Section IV present the methodology of the prediction methods. And the experiment results are drawn in Section V. Section VI make a conclusion.

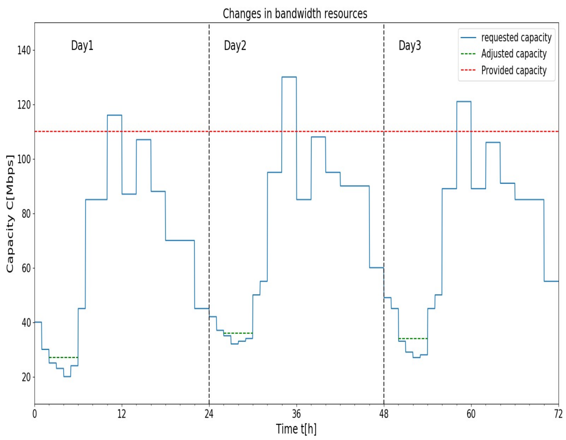

Figure 1. Illustration about application of bandwidth resources prediction for the scenario of dynamic slicing

II. RELATED WORKS

A. Dynamic Network Slicing

To make good use of the physical resources of substrate network and guarantee the quality of experience (QoE) of users, Raza M R *et al.* proposed the concept of dynamic slicing [2]. In their further work [3], Wei F *et al.* assumed the distribution of the bandwidth resources requirements for a slice is known and defined the network slicing reconfiguration problem, which is a sub-task of dynamic slicing, as a MILP, then they employed the reinforcement learning-based approach method to solve it. In [4], the authors conducted their work based on traffic uncertainty, and most of their methods were robust optimization.

B. Network Traffic Prediction

As network traffic prediction is essential for many network management tasks, so it has been a hot topic of research. In [10], Li Y *et al.* firstly decomposed the time series into high-frequency component and low-frequency component by discrete wavelet transform, then they predicted the two components by the Prophet model and Gaussian process regression. In [12], He Q *et al.* proposed a meta-learning scheme to predict a specific kind of network traffic. Their scheme consisted of many kinds of sub-predictors and the main policy to choose the best fit sub-predictors dynamically.

Though there existed many works about network traffic predictions, most of their tasks are short-term (100ms, 5s, 15sec), single-step traffic predictions [11-12], and their evaluation metrics are mean square error (MSE) and mean absolute percentage error (MAPE), these metrics can reflect prediction accuracy but cannot reflect how the prediction result can affect the QoE of slices and the profits of the network providers. What is more, in the scenario of dynamic slicing, we hope to conduct long-term (at least 1 hour), multi-step traffic predictions tasks to determine the adjusted bandwidth resources for the slices. The reason why we do not conduct short-term, single-step traffic predictions is that it is not reasonable to frequently adjust the resources that had been allocated. Currently, few works focus on the bandwidth resources prediction problem of dynamic slicing.

III. DATASET

In this section, we discuss the dataset and the data preprocessing in detail.

A. Dataset Description

The dataset we used is provided by the WIDE project [13]. We monitor the network traffic volume data of a flow that is defined by a source and destination IP prefix in their network for two months (starting from Aug 1, 2021), and we roughly consider that the collected data is the original traffic data of a permanent slice in the network.

B. Dataset Preprocessing

As aforementioned, in the scenario of dynamic slicing, we should conduct the long-term, multi-step traffic prediction task. We count the byte volume of the flow per hour, and then the traffic time series data of a slice can be defined as Eq. (1), where f_i^t denotes the bytes generated by slice i within time slot $[t, t+\tau]$ and in this paper, τ is 1 h. With the definition of time series data, our prediction task is to predict $\{f_i^{t+1},...,f_i^{t+n}\}$ based on its historical time series data F_i, where n is the n_{th} step.

$$F_i = \{f_i^0, f_i^1, ..., f_i^t\} \tag{1}$$

Then, we analyze daily traffic trend for the slice. We define $D = \{f_i^j, f_i^{j+1}, ..., f_i^{j+23}\}\ (j = 0, 24, 48, ...)$ as the daily traffic time series data for slice i, and we draw a traffic trend diagram according to $D_{2021.8.2}$. As presented in Fig. 2, it is obvious that the slice generates much more bytes during the day, which is similar to the conclusion of some research work [5-6]. Then, we employ the Pearson coefficient ρ to verify the similarity of daily traffic trends.

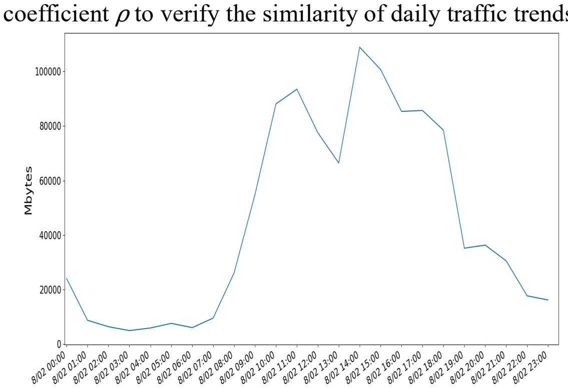

Figure 2. Daily traffic trend of a slice

Pearson coefficient can be calculated according to Eq. (2), where $Cov(X,Y)$ is the covariance between X and Y, and σ_X, σ_Y are the standard deviations of X, Y respectively.

$$\rho = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{2}$$

In our work, X is $D_{2021.8.2}$, Y is D_k. Generally, if $|\rho| > 0.6$, we can consider that X and Y have a strong linear relationship. As presented in Fig. 3, most of the daily traffic time series data are strongly correlated with $D_{2021.8.2}$, that means the diurnal pattern of traffic is a common phenomenon in our dataset.

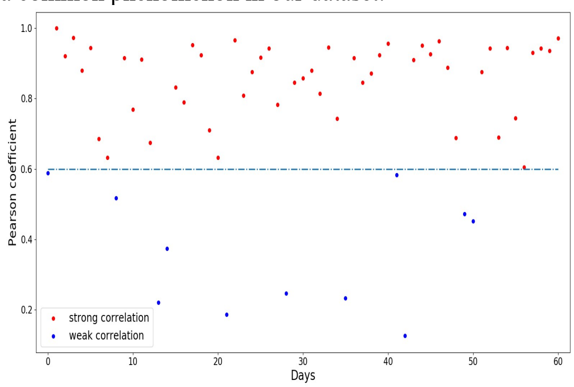

Figure 3. Pearson coefficient between $D_{2021.8.2}$ and other days

Further, we remove the traffic data on weekends, and we find that the weakly correlated points are significantly reduced, which is shown in Fig. 4.

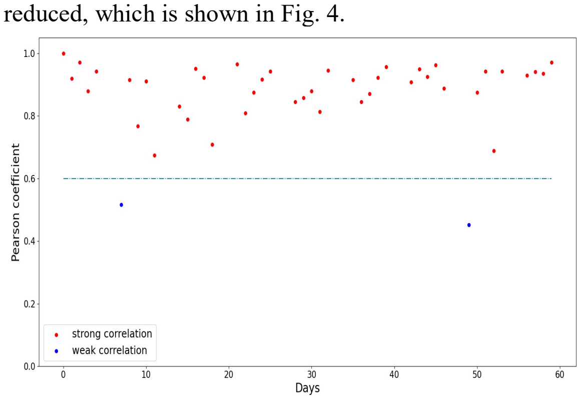

Figure 4. Pearson coefficient between $D_{2021.8.2}$ and other days (skip weekends time series data)

The above analysis shows that in most situations, the daily traffic trends of this slice are similar to $D_{2021.8.2}$. For minimize the number of outlier and make the time series data more stationary, we remove weekend time series data from F_i. And then we only retain the time series data from 2 to 5 o'clock every day to reduce the workload. After processing, time series data can be defined as $F_i' = \{f_i^2, f_i^3, f_i^4, ..., f_i^{t+24}, f_i^{t+1+24}, f_i^{t+2+24}\}$. Now the task is to predict the 2 to 5 o'clock traffic volume in the next day based on F' (in our work, $n = 3$).

IV. Prediction Methods

A. *Autoregressive Integrated Moving Average (ARIMA)*

ARIMA is a classical model and it is widely used for time series data analysis [7]. Typically, ARIMA consists of three elements, autoregression, integrated, and moving average, the detail of these elements can be found in [7].

An ARIMA model can be defined as Eq. (3), where Δy_t^d is the d_{th} differenced variable and it should be stationary, c is constant, the terms in ϕ_i are autocorrelation coefficients at lags 1, 2,..., p, ε_t is a Gaussian white noise series, and the θ_i terms are the weights applied to the current and previous values of a stochastic term in the time series.

$$\Delta y_t^d = c + \sum_{i=1}^{p} \phi_i \Delta y_{t-i}^d + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{3}$$

The most important capacity of ARIMA model is that it can analyze the time series data which is non-stationary by means of the integrate step. To determine parameter *d* of the ARIMA model, we perform first-order difference processing on F', and then use Augmented Dickey-Fuller (ADF) test to verify the stationarity of it. The *p* value is $4.5729616*10^{-13}$, which is significantly less than the baseline (0.05), so we can consider the processed time series data is stationary, and set *d* to 1.

In order to determine the other parameters *p, q* effectively, similar to the method in [12], we also use the autocorrelation function (ACF) and partial autocorrelation function) (PACF) of our data to analyze autocorrelation of the time series. As presented in Fig. 5 and Fig. 6, we can easily determine the optimal model parameters.

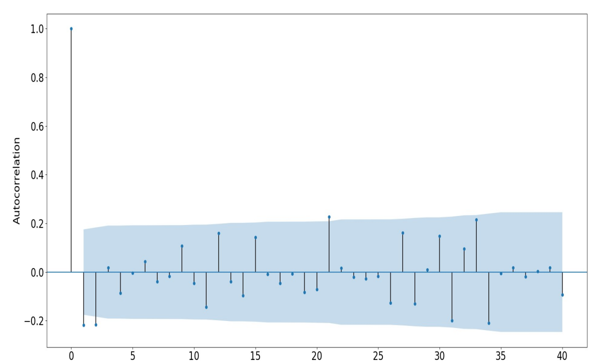

Figure 5. Autocorrelation Function (ACF)

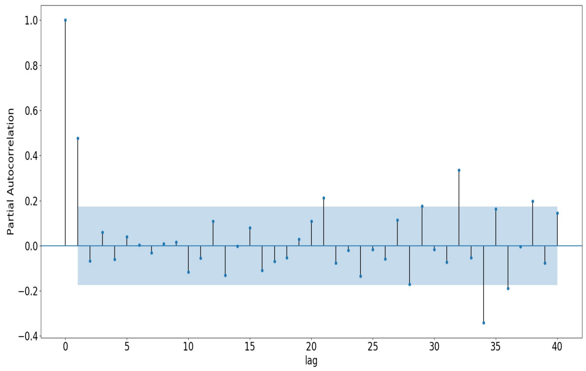

Figure 6. Partial Autocorrelation Function (PACF)

B. Long Short-Term Memory (LSTM) Neural Network

As a form of the recurrent neural network, the powerful capacity in analyzing time series data of LSTM [8] has been proven. So, we evaluate the prediction performance of LSTM in our dataset.

A LSTM cell unit consists of a cell, an input gate, an output gate, and a forget gate. Generally, LSTM models can be expressed by the following equations.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

$$f_t = \sigma(W_f x_t + U_o h_{t-1} + b_f) \tag{6}$$

$$c_t = f_t * c_{t-1} + i_t * \sigma(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$h_t = o_t * \sigma(c_t) \tag{8}$$

In the above equations, i, o, f, c, h are the input, output, forget gates, current state, and hidden state respectively. W represents the input weights of the gates, while U represents the recurrent weights of the gates, and b is the bias vector. A detailed introduction of LSTM model can be found in [8].

In our work, we design our LSTM neural network referred to some existing works [12], and turn prediction problems into supervised learning.

C. Prophet

Prophet is an effective method for time series data analysis. In fact, Prophet takes the prediction problems as a curve-fitting issue, it comprehensively considers the trend, seasonality, and holiday factors of the time series data. The Prophet model can be defined as Eq. (9), where $m(t)$ is a trend function that can model the non-periodic component of the time series data, $s(t)$ represents the periodic component, and $e(t)$ represents the affect of holidays. Finally, ϵ_t is the remaining components that are cannot represented by the above three functions. Typically, we assume that ϵ_t is normally distributed. The detailed calculation progress of these components can be found in [9].

$$y(t) = m(t) + s(t) + e(t) + \epsilon_t \tag{9}$$

It is worth noting that Prophet can perform the task of multi-step interval prediction, and it can independently analyze the characteristics of time series data, so we take F_i as the input directly and take the upper bound of the prediction interval as the prediction result.

V. EXPERIMENTS

A. Evaluation metrics

To measure the prediction result objectively, we define two metrics *P, Q* that can be calculated according to Eq. (10) and Eq. (11). Here, $Predict_{max}$ is the maximum value for one multi-step prediction task, $Actual_{max}$ is the maximal bandwidth resources requirements of a slice from 2 to 5 o'clock on a day. $Allocate_{max}$ is the maximal bandwidth resources that the network providers allocate to a slice on a day, currently, we set $Allocate_{max}$ directly to D_{kmax}.

$$P = \frac{Predict_{max}}{Actual_{max}} \tag{10}$$

$$Q = \frac{Allocate_{max}}{Predict_{max}} \tag{11}$$

As aforementioned, P is applied to measure the extent to which the prediction results meet the bandwidth resources requirements of a slice, Q is applied to measure the extent to which the predicted value can save bandwidth resources. Further, we define a value *tol* to represent the sensitivity about bandwidth resources of services, as shown in Table I. Once $P < tol$, we consider that the prediction result will cause QoE violation.

TABLE I. SENSITIVITY ABOUT BANDWIDTH RESOURCES OF SERVICES

Service	Video conference	HD video	Content cache
tol	1	0.8	0.6

B. Evaluation

In our work, we retain the time series data for the last five weekdays in our dataset for testing the prediction performance of these methods, while other data are used to train the models. The prediction result is shown in Fig. 7.

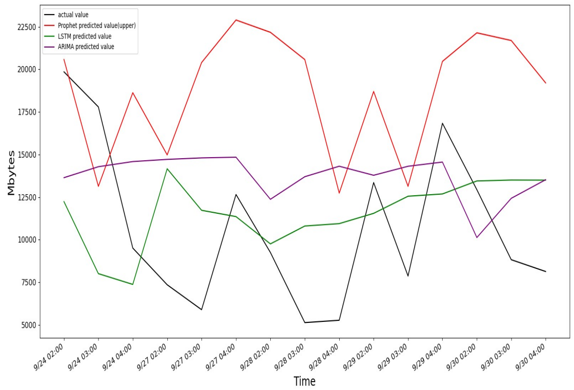

Figure 7. prediction results of different methods

But this figure is hard to reflect how the prediction result can affect the QoE of slices and the profits of network providers. So, we evaluate the predicted values based on *P*, *Q*, and the results are shown in Table II.

TABLE II. ANALYSIS OF PREDICTION RESULTS OF DIFFERENT METHODS

	Day1		Day2		Day3		Day4		Day5	
	P	*Q*	*P*	*Q*	*P*	*Q*	*P*	*Q*	*P*	*Q*
Prophet (upper)	**1.1**	5.1	**1.8**	3.8	**2.5**	4.7	**1.2**	4.5	**1.7**	4.4
ARIMA	0.7	7.2	1.2	5.8	1.6	7.4	0.9	6.4	1.1	**6.9**
LSTM	0.6	**8.6**	1.1	**6.1**	1.2	**9.7**	0.8	**7.3**	1.1	**6.9**

According to the above table, we can intuitively find that Prophet can guarantee the QoE of slices to the greatest extent, while LSTM can save more bandwidth resources for network providers and may cause QoE violations. ARIMA can consider the network slices' QoE and the network providers' revenue in a more balanced way. In the scenario of dynamic network slicing, network operator can select prediction method according to the slices' service type. For services with high bandwidth requirements ($tol = 1$) such as video conferencing and remote surgery, operators need to use Prophet to predict their bandwidth resources demand to ensure their service quality. For services such as HD video that have a certain tolerance for bandwidth ($tol \in [0.8,1)$), operators can use ARIMA to predict their bandwidth resources demand, so as to ensure their quality of service and recover some bandwidth resources at the same time. For bandwidth insensitive services ($tol < 0.8$) such as Content cache, operators can use LSTM to predict their bandwidth requirements in order to recover bandwidth resources as much as possible.

VI. CONCLUSION AND FUTURE WORK

In this paper, we employ the time series analysis theory to analyze the network traffic data, and evaluate the perform of the state-of-art time series prediction methods in the scenario of dynamic slicing. Further, we use two novel evaluation metrics to reflect how the prediction results can affect the users and the network providers. In the future, we will investigate how to improve the prediction performance through some other deep learning methods, and conduct some works related to network slicing reconfiguration based on these prediction methods.

ACKNOWLEDGMENT

This research work was supported in part by the Fujian Province Natural Science Foundation of China under Agreement 2020J01574 and Industry-university-research Innovation Fund of China under Agreement 2021FNA05003.

REFERENCES

[1] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, (2017) Network slicing for 5G: Challenges and opportunities. IEEE Internet Computing, 21(5): 20-27.

[2] M. R. Raza, M. Fiorani, A. Rostami, P. Öhlen, L. Wosinska, and P. Monti. (2017) Benefits of programmability in 5G transport networks. In: Optical Fiber Communication Conference. Los Angeles. M2G-3

[3] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin, and Y. C. Liang. (2020) Network slice reconfiguration by exploiting deep reinforcement learning with large action space. IEEE Transactions on Network and Service Management, 17(4): 2197-2211.

[4] A. Baumgartner, T. Bauschert, A. A. Blzarour, and V. S. Reddy. (2017) Network slice embedding under traffic uncertainties — A light robust approach. In: 13th International Conference on Network and Service Management. Tokyo. pp. 1–5.

[5] T. Benson, A. Akella, and D. A. Maltz. (2010) Network traffic characteristics of data centers in the wild. In: Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. New York. pp. 267-280.

[6] G. Dan and N. Carlsson. (2014) Dynamic content allocation for cloud-assisted service of periodic workloads. In: IEEE INFOCOM 2014-IEEE Conference on Computer. Toronto. pp. 853–861.

[7] T. W. Anderson. (1994) The Statistical Analysis of Time Series. Wiley, New Jersey.

[8] S. Hochreiter, J. Schmidhuber. (1997) Long short-term memory. Neural computation, 9(8): 1735-1780.

[9] S. J. Taylor, B. Letham B. (2018) Forecasting at scale. The American Statistician, 72(1): 37-45.

[10] Y. Li, Z. Ma, Z. Pan, N. Liu, and X. You. (2020) Prophet model and Gaussian process regression based user traffic prediction in wireless networks. Science China Information Sciences, 63(4): 1-8.

[11] A. Lazaris, and V. K. Prasanna. (2017) DeepFlow: a deep learning framework for software-defined measurement. In: Proceedings of the 2nd Workshop on Cloud-Assisted Networking. Incheon. pp. 43-48

[12] Q. He, A. Moayyedi, G. Dán, G. P. Koudouridis, and P. Tengkvist. (2020) A meta-learning scheme for adaptive short-term network traffic prediction. IEEE Journal on Selected Areas in Communications, 38(10): 2271-2283.

[13] K. Cho, K. Mitsuya, and A. Kato. (2000) Traffic Data Repository at the WIDE Project. In: Proceedings of the annual conference on USENIX Annual Technical Conference. California. pp. 51-51.